An EM Algorithm for Nonlinear State Estimation with Model Uncertainties

Amin Zia, Thia Kirubarajan, James P. Reilly*, Derek Yee, Kumaradevan Punithakumar, and Shahram Shirani

Department of Electrical and Computer Engineering

McMaster University

1280 Main Street West,

Hamilton, Ontario, Canada L8S 4K1

Email: {ziama, kiruba, reillyj, shirani}@mcmaster.ca

phone: 905 525 9140 x22895 (corresponding author)

fax: 905 521 2922

Abstract

In most solutions to state estimation problems, e.g., target tracking, it is generally assumed that the state transition and measurement models are known a priori. However, there are situations where the model parameters or the model structure itself are not known a priori or are known only partially. In these scenarios, standard estimation algorithms like the Kalman filter and the extended Kalman Filter (EKF), which assume perfect knowledge of the model parameters, are not accurate. In this paper, the nonlinear state estimation problem with possibly non-Gaussian process noise in the presence of a certain class of measurement model uncertainty is considered. It is shown that the problem can be considered as a special case of maximum likelihood estimation with incomplete-data. Thus, in this paper, we propose an EM-type algorithm that casts the problem in a joint state estimation and model parameter identification framework. The expectation (E) step is implemented by a particle filter that is initialized by a Monte-Carlo Markov chain algorithm. Within this step, the posterior distribution of the states given the measurements, as well as the state vector itself, are estimated. Consequently, in the maximization (M) step, we approximate the nonlinear observation equation as a mixture of Gaussians (MoG) model. During the M-step, the MoG model is fit to the observed data by estimating a set of MoG parameters. The proposed procedure, called EM-PF (*expectation–maximization particle filter*) algorithm, is used to solve a highly nonlinear bearing-only tracking problem, where the model structure is assumed unknown *a priori*. It is shown that the algorithm is capable of modelling the

*Corresponding author.

observations and accurately tracking the state vector. Additionally, the algorithm is also applied to the sensor registration problem in a multi–sensor fusion scenario. It is again shown that the algorithm is successful in accommodating an unknown nonlinear model for a target tracking scenario.

Indexing Terms: Nonlinear estimation, system identification, expectation-maximization algorithm, MCMC, particle filters, nonlinear regression, joint estimation-identification, sensor fusion, sensor registration, sensor bias estimation.

1. INTRODUCTION

In most solutions to state estimation problems, both linear and nonlinear, it is generally assumed that the state transition process and the measurement process parameters are known a priori. For instance, in target tracking using a nonlinear state space model, the extended Kalman filter (EKF) assumes that the process and measurement matrices as well as corresponding noise statistics are known [1]. However, there are situations in which the model parameters are not known a priori or they are only known with some degree of uncertainty. In these state estimation problems neither a set of certain training data is available to accurately identify the model uncertainties, nor are accurate models of the measurement process available to precisely estimate the states using conventional estimation procedures. In these circumstances, standard estimation algorithms, which are based on perfect knowledge of the model parameters, are not accurate anymore. To solve this problem, we must perform optimum state estimation in the presence of model uncertainty; i.e., perform model identification while tracking.

There are three main categories of methods proposed to perform this task. The classical remedy is to treat the unknown parameters as extra state variables and augment the state vector by the unknown parameters [1]. For a review of these methods refer to [2]. See also the reduced state estimator approach [3].

The second category consists of the so-called multiple model (MM) estimators [1]. An adaptive filtering algorithm decides the most appropriate model from a number of different but predefined model dynamics during the estimation process. The generalized pseudo-Bayesian estimator (GSBS) and the interactive multiple model estimation (IMM) procedure are among the best known examples [1] of this type of method. The MM estimators with variable structure are another example of this type of estimator. For a comprehensive review of these methods, interested readers are referred to the survey paper [4] along with [5]–[7]. MM algorithms show promising performance in tracking maneuvering targets whose dynamics are predictable. However, their ability to handle model uncertainties is limited to the “model dictionary” available. Furthermore, they often require a long time to acquire track, as shown in the simulation section of this paper.

The key idea in the third category of algorithms is to divide the problem of state estimation in the presence of the model uncertainty into two joint problems; i.e., state estimation and model identification [8]. Firstly, assuming that the model is known perfectly, the states are estimated. Then the estimated states with their corresponding measurements are used to identify the model parameters. Perhaps the first paper in this regard was [9]. Also, in [2], an optimality test was derived to adjust a Kalman filter when the noise statistics are not known exactly. Later in [10] and [11], joint simultaneous state estimation and model identification for the scalar state estimation case, in the presence of unknown model parameters, was studied. See also [12]–[16] for similar approaches.

This third category is the one chosen for this paper. Interestingly, this approach can be cast in an expectation maximization (EM) context [17]. See [18] in which a general framework for solving the general joint estimation-identification of *linear Gaussian* models was presented. Refer to [19], [20] for a similar application of the EM algorithm for linear state estimation with uncertain model parameters. The main idea in EM-based algorithms is to solve the state estimation problem in the presence of model uncertainty in two iterative steps. In the first step, called the E-step, it is assumed that the model is known perfectly and therefore standard estimation methods are used to estimate the states. Then, in the second step, i.e., the M-step, the estimated states with their corresponding measurements are used to identify the model parameters. Different implementations of the E and the M steps have resulted in different algorithms suitable for different applications.

In this paper we extend the approaches of [18] and others with regard to the problem of model identification while tracking. Here, we extend the previous work to the case where the measurement model is nonlinear and unknown, subject to the restriction that it can be accurately represented as a mixture of Gaussian (MoG) kernels, and that the Cramer–Rao bound for all the model parameters and the states, given the observations, exists. A specific EM procedure called the EM-PF algorithm [21], is presented.

In the E-step of the proposed algorithm, an approximation of the posterior distribution of the states given the measurements is formulated. This distribution is then used to estimate the states. In nonlinear systems this conditional density is generally non-Gaussian and can be quite complex. We use a particle filter [22] algorithm to estimate and recursively update this posterior distribution in time. Because the EM algorithm is sensitive to initialization, the particle filter is initialized using a Metropolis-Hastings Monte-Carlo Markov Chain (MH-MCMC) [22] procedure. This greatly assists the algorithm in converging to the global optimum. In the maximization (M) step, the unknown measurement process is approximated by fitting the observations to an MoG model using the current estimate of the states. A closed–form maximum likelihood procedure for determining the parameters of the MoG model is

given.

Finally, the proposed EM-PF algorithm is applied to two nonlinear state estimation problems with model uncertainties. First, we consider a typical bearing-only tracking problem where the sensors have an unknown measurement bias. In this example we treat the observation model in the presence of sensor biases as unknown. It is shown that the EM-PF algorithm is capable of successfully estimating the position and velocity states and therefore can accommodate model uncertainty and correct the misalignment caused by the sensor bias. Then, we approach a sensor registration problem in which different sensors with different unknown bias values combine their measurements for state estimation. Here again we treat the observation model as unknown. We show that the sensor registration is performed successfully and the effect of sensor bias is suppressed by the algorithm. Even though in each of the above examples it may be possible to gain better performance by exploiting the known form of the nonlinear model, we demonstrate that the proposed method is applicable to situations where very little is known about the structure of the observation model.

The structure of paper is as follows. In Section 2, the general framework for the EM algorithm is introduced. The details of the proposed EM–PF algorithm follow. The implementation of the E-step using a particle filter, and the M-step by fitting an MoG model to the estimated data, are provided in Section 3. Then in Section 4, the proposed method is applied to a nonlinear bearing-only tracking problem (similar to the one in [23]) with uncertain model parameters. Also, Section 5 presents the application of the EM-PF algorithm to a sensor-registration problem in a multisensor tracking scenario. Simulation results are presented for each application. Concluding remarks close the paper.

An upper–case bold symbol (e.g., $\boldsymbol{A}$) denotes a matrix, and a lower–case bold symbol denotes a vector. If the vector is a function of time, e.g., $\boldsymbol{z}(t)$, then the corresponding symbol without the time index (e.g., $\boldsymbol{z}$) denotes the set of all values of the vector over the range of the temporal index; e.g., $\boldsymbol{z}$ denotes $\{\boldsymbol{z}(t)|t = 1, \ldots, L\}$, where L is the number of data points. The notation $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ indicates a Gaussian distribution with mean $\boldsymbol{m}$ and covariance $\boldsymbol{\Sigma}$.

Throughout the paper, t where $t = 1, \ldots, L$ denotes the *discrete* time index, $k = 1, 2 \ldots$ is the EM iteration index where $k = 1$ is the initialization step, and i, where $i = 1, \ldots, N$ is the particle index, where N is the number of particles used in the particle filter.

2. Nonlinear State Estimation using EM

State estimation in a nonlinear state-space dynamical system whose evolution process is described as

$$\boldsymbol{x}(t+1) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{u}(t), \tag{1}$$

consists of estimating the state data vector $\boldsymbol{x}$ using a sequence of noisy measurements given by the following model:

$$\boldsymbol{z}(t) = \boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big) + \boldsymbol{v}(t), \qquad t = 1, 2, \ldots, \tag{2}$$

where t is the *discrete* time index, $\boldsymbol{x}(t) \in \mathbb{C}^M$ and $\boldsymbol{z}(t) \in \mathbb{C}^J$ are the state variable and the noisy output measurement vectors respectively, and $\boldsymbol{u}(t) \in \mathbb{C}^M$ is assumed to be an *i.i.d* noise processes, whose probability density function is assumed known and possibly non–Gaussian. The vector $\boldsymbol{v}(t) \in \mathbb{C}^J$ is a zero-mean Gaussian noise variable with unknown covariance $\boldsymbol{Q}$. The noise $\boldsymbol{v}(t)$ is assumed uncorrelated in time; i.e., $E\big(\boldsymbol{v}(t_1)\boldsymbol{v}(t_2)^H\big) = \Delta(t_1 - t_2)\boldsymbol{Q}$, where

$$\Delta(t) = \begin{cases} 1, & t = 0; \\ 0 \text{ otherwise.} \end{cases}$$

Also, the vector valued functions $\boldsymbol{f}, \quad \boldsymbol{f} : \mathbb{C}^M \mapsto \mathbb{C}^M$, and $\boldsymbol{h}, \quad \boldsymbol{h} : \mathbb{C}^M \mapsto \mathbb{C}^J$ are assumed to be smooth but otherwise are arbitrary. We assume that the function $\boldsymbol{f}(\cdot)$ is known, whereas uncertainty may exist in the observation model $\boldsymbol{h}(\cdot)$.

A major focus of this paper is how to model the partially known or unknown function $\boldsymbol{h}(\cdot)$. If a model which takes into account any known structure in the measurement process is available, then that model should be used in the proposed method. Any uncertainty in $\boldsymbol{h}(\cdot)$ is expressed in a parameter vector $\boldsymbol{\theta}$, which is modelled as an unknown deterministic vector. On the other hand, it is also possible to assume no structure on $\boldsymbol{h}(\cdot)$, as is done with our examples in Sects. 4 and 5. We model this function as an MoG, again parameterized by the vector $\boldsymbol{\theta}$, in a manner to be described later in Sect. 3-B.

An obvious restriction so that the proposed methodology can yield useful results is that states $\boldsymbol{x}$ must be observable from the observations, in the presence of the unknown parameters $\boldsymbol{\theta}$ describing the model $\boldsymbol{h}(\cdot)$. That is, the joint Cramer–Rao bound on the states and the unknown parameters $\boldsymbol{\theta}$ given the observations, must have finite values for the states. Here we do not discuss conditions for which the bound exists. However, it is clear that the proposed formulation will place restrictions on the class of problems that may be considered.

When the model is known completely, maximum likelihood (ML) state estimation results in a filtering problem, which can be solved using, e.g., the EKF, the particle filter, or the unscented Kalman filter [24]. Also, in the case where when the model structure is known but contains a number of unknown parameters, and a training set consisting of corresponding state and measurement data is available, then the states and unknown model parameters can be jointly estimated using maximum likelihood (ML) procedures, as is common practice in communication systems. However, in the case considered here where we assume that no training set is available, and the measurement function $\boldsymbol{h}$ is uncertain or unknown, standard estimation algorithms that assume perfect knowledge of the model parameters are

not accurate. In this case it is desirable to jointly estimate the state vectors and the observation model using an EM technique which blindly incorporates model uncertainty, as is proposed in this paper.

To estimate the states in the presence of model uncertainty, we use the *variational* form of the EM algorithm [18]. The log likelihood of observations is defined as:

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\boldsymbol{z}|\boldsymbol{\theta}) = \log \int_{\chi} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} \tag{3}$$

where χ is the range of the state variables, $\boldsymbol{z} = [\boldsymbol{z}^T(1), ..., \boldsymbol{z}^T(L)]^T \in \mathbb{C}^{JL}$ is the entire sequence of observed measurements, $\boldsymbol{x} = [\boldsymbol{x}^T(1), ..., \boldsymbol{x}^T(L)]^T \in \mathbb{C}^{ML}$ are all the state variables, $\boldsymbol{\theta}$ is the vector of parameters describing the MoG model, and L is the number of observation points.

Maximizing this function can often be intractable in the nonlinear/non-Gaussian case. Therefore, an alternative procedure is to define a variational distribution $U(\boldsymbol{x})$ over the hidden state variables, that allows us to obtain a lower bound on the expected likelihood [18], [25]:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= \log \int_{\chi} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} \\
&= \log \int_{\chi} U(\boldsymbol{x}) \frac{\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{U(\boldsymbol{x})} d\boldsymbol{x} \\
&\geq \int_{\chi} U(\boldsymbol{x}) \log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})}{U(\boldsymbol{x})} d\boldsymbol{x} \qquad (4) \\
&= \int_{\chi} U(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} - \int_{\chi} U(\boldsymbol{x}) \log U(\boldsymbol{x}) d\boldsymbol{x} \\
&= \int_{\chi} U(\boldsymbol{x}) \log p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x} + H(U) \qquad (5) \\
&\triangleq \mathcal{F}(U, \boldsymbol{\theta}),
\end{aligned}$$

where (4) follows from Jensen's inequality [1] and $H(U)$ in (5) is the entropy of the distribution U. It is straightforward to show [18] that the equality in (4) is satisfied for $U^{\star}(x) = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$.

The EM algorithm alternates between maximizing $\mathcal{F}$ with respect to the distribution $U(\boldsymbol{x})$ and the parameters $\boldsymbol{\theta}$, respectively. Starting from some initial parameters $\boldsymbol{\theta}_0$ the algorithm iteratively applies

$$E - Step: \quad U_{k+1} = \arg\max_{U} \mathcal{F}(U_k, \boldsymbol{\theta}_k) \tag{6}$$

$$M - Step: \quad \boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \mathcal{F}(U_{k+1}, \boldsymbol{\theta}_k), \tag{7}$$

where k is the EM iteration index. The primary purpose of the the E-step is to estimate the hidden states. This is accomplished by determining the best distribution $U^{\star} = p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})$ which makes the expectation of log-likelihood maximum. A conditional–mean estimate of the states is then readily available from this distribution. The M-step involves estimating the model parameters $\boldsymbol{\theta}$ using the states estimated in the previous E-step and their corresponding measurements. Since, at the end of

each E-step the likelihood function $\mathcal{F}$ meets the equality for $U^\star(x)$, then $\mathcal{F}(U^*_{k+1}, \boldsymbol{\theta}_k) = \mathcal{L}(\boldsymbol{\theta}_k)$. Also, because in the M-step the optimization is over $\boldsymbol{\theta}$, it is guaranteed that the likelihood will not decrease in any iteration.

3. The EM–PF Algorithm

The overall operation of the proposed EM–PF algorithm for estimating states in the presence of model uncertainties, nonlinear models and non–Gaussian noise is shown in Fig. 1. The algorithm as shown in this figure operates in batch mode, using a finite set of observations $\boldsymbol{z}(t), t = 1, \ldots, L$. Since we wish to estimate the states $\boldsymbol{x}_k(t)$ over this same interval, the problem may be cast as a fixed interval smoothing problem. It is assumed that the parameters $\boldsymbol{\theta}_k$ describing the model do not change significantly over this interval. In the situation of interest in this paper, where the observation noise is non–Gaussian or the model is nonlinear, the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, which is critical to the E–step, cannot be evaluated analytically. In the proposed EM-PF algorithm, this distribution is approximated using a particle filter.

A. The E-step: Estimation of States by the Particle Filter

At the kth iteration of the EM algorithm, the distribution of interest for the E–step is $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, for $t = 1, \ldots, L$. When the noise is non–Gaussian or the model is nonlinear, this distribution can be intractable. Instead, an approximation $\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}, \boldsymbol{\theta}_k)$ to this distribution is used, which is propagated in time by a particle filter. At the beginning of the kth E-step, it is assumed that the parameter vector $\boldsymbol{\theta}_k$ has been estimated within the previous M-step and therefore is known.

Given $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$, the states $\boldsymbol{x}_k(t)$ can be estimated as, e.g., the conditional mean of this distribution at any time $t = 1, \ldots, L$. Then in the M-step, the estimated states and their corresponding measurements are used to identify the measurement function $h()$, parameterized by $\boldsymbol{\theta}_k$. This vector is estimated using the state-measurement pairs estimated in the E-step. The E– and M–steps iterate until convergence.

The presentation on particle filters here is necessarily brief; readers are referred to [22], [26] for further background. We first explain the case for the *filtering* distribution; i.e., approximation of the filtering distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t}, \boldsymbol{\theta}_k)$[1]. We later extend the treatment to the *fixed–interval smoothing* problem, i.e., approximation of the distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k)$, for any $t \in [1, \ldots, L]$, which is the problem of relevance for this paper.

[1]The notation $\boldsymbol{y}_{a:b}$ is commonly used in the particle filtering literature and implies all values of $\boldsymbol{y}$ from time a to b.

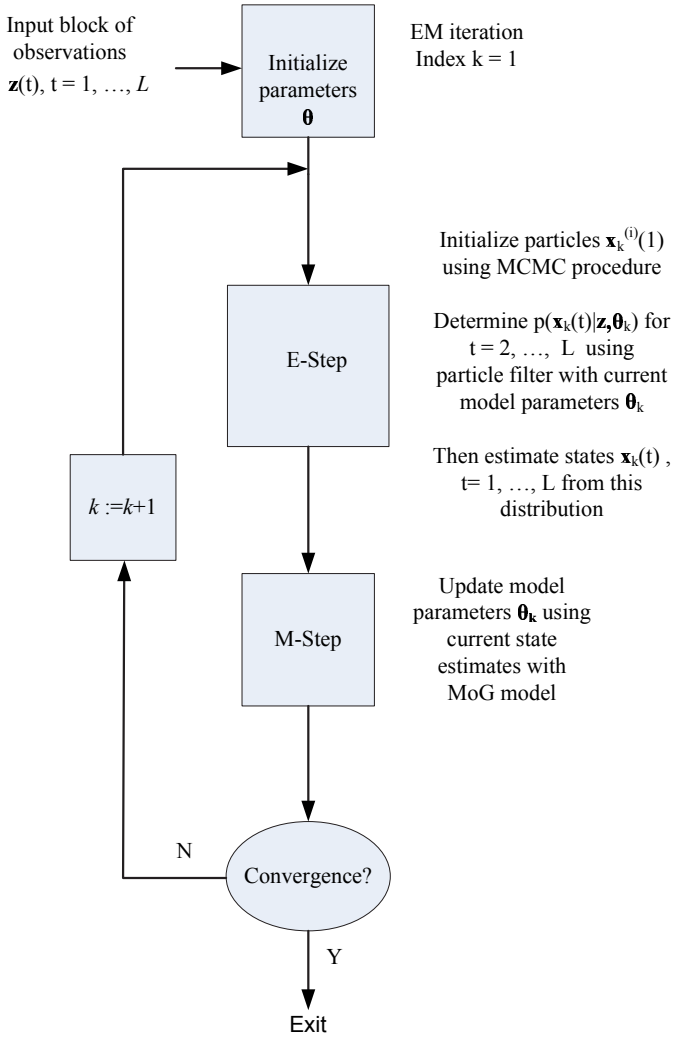


Fig. 1. Block diagram of the EM–PF algorithm, which gives state and model parameter estimates, over the block $t = 1, \ldots, L$.

The quantity $\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t},\boldsymbol{\theta}_k)$ is specified by a set $\big\{\boldsymbol{x}_k^{(i)}(t),w_k^{(i)}(t)\big\}_{i=1:N}$, where the $\boldsymbol{x}_k^{(i)}(t)$ are samples (particles) of the states, that are used to compose the desired distribution. The quantity N is the number of particles, and $w_k^{(i)}(t)$ are the respective filtering weights, whose calculation is described below. The approximation $\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t},\boldsymbol{\theta}_k)$ is given by

$$\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t},\boldsymbol{\theta}_k)=\sum_{i=1}^{N}w_k^{(i)}(t)\delta\big(\boldsymbol{x}_k(t)-\boldsymbol{x}_k^{(i)}(t)\big), \tag{8}$$

where $\delta(\cdot)$ is the Dirac delta function.

The unnormalized weights at time t can be recursively updated from those at time $t-1$ at EM iteration k by [22], [26]

$$\tilde{w}_k^{(i)}(t)=w_k^{(i)}(t-1)\frac{p(\boldsymbol{z}(t)|\boldsymbol{x}_k^{(i)}(t),\boldsymbol{\theta}_k)p(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1))}{r(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1),\boldsymbol{z}(t))},\qquad i=1,\ldots,N. \tag{9}$$

The normalized weights $w_k^{(i)}(t)$ are then calculated as

$$w_k^{(i)}(t)=\frac{\tilde{w}_k^{(i)}(t)}{\sum_{i=1}^{N}\tilde{w}_k^{(i)}(t)},\qquad i=1,\ldots,N. \tag{10}$$

The quantities $\boldsymbol{x}_k^{(i)}(t)$ in (9) are the particles, which are samples drawn from a proposal distribution $r\big(\boldsymbol{x}_k^{(i)}(t)\big|\boldsymbol{x}_k^{(i)}(t-1)\big)$. This distribution is chosen to be easy to sample from, and to resemble the desired distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z},\boldsymbol{\theta}_k)$ as closely as possible. In this vein we choose the proposal distribution to be a normal distribution:

$$r(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1))\sim\mathcal{N}(\boldsymbol{x}_k(t-1),\sigma_r^2\boldsymbol{I}), \tag{11}$$

where σ_r^2 is chosen to give the best fit to $r(\cdot|\cdot)$.

The distribution $p(\boldsymbol{z}(t)|\boldsymbol{x}_k^{(i)}(t),\boldsymbol{\theta}_k)$ in (9) is the likelihood, and is determined from (2), given $\boldsymbol{\theta}_k$ and knowledge of the distribution of $\boldsymbol{v}(t)$. The distribution $p(\boldsymbol{x}_k^{(i)}(t)|\boldsymbol{x}_k^{(i)}(t-1))$ in (9) is the prior distribution on the states and is given from (1), knowing the distribution of $\boldsymbol{u}(t)$. Thus, the method propagates the desired distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:t},\boldsymbol{\theta}_k)$ in time at each value of t by first, drawing particles $\boldsymbol{x}_k^{(i)}(t),i=1,\ldots,N$ from the proposal distribution $r(\cdot|\cdot)$. Then, using the particles and the observations, the respective distributions in (9) can be evaluated. The weights $w_k^{(i)}(t)$ are then updated using (9) and (10), whereupon the desired approximate distribution is obtained by (8).

We now extend this treatment to the fixed–interval smoothing problem. It is shown in [27] that the smoothing distribution is given as

$$\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:L},\boldsymbol{\theta}_k)=\sum_{i=1}^{N}w_k(t|L)^{(i)}\delta(\boldsymbol{x}_k(t)-\boldsymbol{x}_k^{(i)}(t)) \tag{12}$$

for any $t \in [1, \ldots, L]$. Thus, only the weights change in going from the filtering to the smoothing problem. The smoothing weights $w_k^{(i)}(t|L)$ are calculated according to the following algorithm [27]:

1) Initialization at time $t = L$:
   - for $i = 1, \ldots, N$, $w_k^{(i)}(L|L) = w_k^{(i)}(L)$.
2) for $t = L, \ldots, 1$
   - for $i = 1, \ldots, N$, evaluate the smoothing weights:

$$w_k^{(i)}(t|L) = \sum_{m=1}^{N} w_k^{(m)}(t+1|L) \frac{w_k^{(i)}(t) p(\boldsymbol{x}^{(m)}(t+1)\big|\boldsymbol{x}^{(i)}(t))}{\left[\sum_{\ell=1}^{N} w_k^{(\ell)}(t) p\big(\boldsymbol{x}^{(m)}(t+1)\big|\boldsymbol{x}^{(\ell)}(t)\big)\right]}. \tag{13}$$

Using (12), conditional mean state estimates $\hat{\boldsymbol{x}}_k(t)$ can be obtained for any time $t = 1, \ldots, L$ by

$$\begin{aligned} \hat{\boldsymbol{x}}_k(t) &= \int_{\mathcal{X}} \boldsymbol{x}_k(t) p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k) d\boldsymbol{x}_k(t) \\ &\approx \int_{\mathcal{X}} \boldsymbol{x}_k(t) \hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}_{1:L}, \boldsymbol{\theta}_k) d\boldsymbol{x}_k(t) \\ &= \sum_{i=1}^{N} w_k^{(i)}(t|L) \boldsymbol{x}_k^{(i)}(t). \end{aligned} \tag{14}$$

In the sequel, for ease of notation we write $\boldsymbol{z}$, implying $\boldsymbol{z}_{1:L}$

The problem with the particle filter is that after a few time steps, all but a very few of the particles have negligible weights. This degeneracy problem results in inefficient use of the particles. There are a number of proposed resampling techniques that correct this problem. A simple minimum variance scheme first proposed by Kitagawa [28], and applied to a tracking problem [23], is used in this paper. This re-sampling technique probabilistically replicates particles with large weights and discards particles with small weights, so that our set of particles better represents the required distribution.

Initialization of the particle filter at $t = 1$: The initial particles at time $t = 1$ for each EM iteration k must be chosen carefully, otherwise the particle filter may lose track later in time. For this purpose, we consider the Metropolis-Hastings (MH) algorithm, which is a Monte Carlo Markov chain (MCMC) procedure, for generating samples from the initial posterior distribution $\pi \stackrel{\Delta}{=} p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$. Ideally, we would like to use the exact distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{z}, \boldsymbol{\theta}_k)$; however, this is not possible for reasons of tractability, so we use π as an approximation. As described below, the MCMC process is iterative; each iteration places an underlying Markov chain in a different state, which corresponds to a sample; thus, a potential candidate sample is drawn in each iteration. An appropriate number of initial iterations (referred to as the *burn–in* period), are required before the underlying Markov chain establishes equilibrium. Only after equilibrium is established are the samples distributed according to the desired distribution π; therefore, the burn–in samples are discarded. After the burn-in period completes, N useful samples are drawn by executing N additional iterations. These additional samples serve as the

initial particles $x_k^{(i)}(1)$ for the particle filter. Since these initial particles are already distributed according to the approximate desired posterior distribution, the corresponding weights are all initialized to unity.

By choosing a proposal density $q(\boldsymbol{x}|\cdot)$ which may be different from $r(\cdot|\cdot)$, the following procedure generates samples, $\boldsymbol{x}_k^{(i)}(1)$ from $\pi = p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$:

for $i = 1, \ldots N$, after equilibrium of π is reached:

1) Sample $\boldsymbol{x}^* \sim q(\boldsymbol{x}|\boldsymbol{x}^{(i-1)})$

2) Evaluate

$$\alpha\big(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^*\big) \triangleq \min\left\{1, \frac{\pi(\boldsymbol{x}^*)q(\boldsymbol{x}^{(i-1)}|\boldsymbol{x}^*)}{\pi(\boldsymbol{x}^{(i-1)})q(\boldsymbol{x}^*|\boldsymbol{x}^{(i-1)})}\right\} \tag{15}$$

3) assign $\boldsymbol{x}_k^{(i)}(1) = \boldsymbol{x}^*$ with probability $\alpha(\boldsymbol{x}^{(i-1)}, \boldsymbol{x}^*)$.

We choose the proposal density $q(\boldsymbol{x}|\boldsymbol{x}^{(i-1)})$ to be an easy-to-sample distribution; e.g., the normal distribution:

$$q(\boldsymbol{x}|\boldsymbol{x}^{(i-1)}) \sim \mathcal{N}(\boldsymbol{x}^{(i-1)}, \sigma_q^2 \boldsymbol{I}), \tag{16}$$

where, in this study, the variance σ_q^2 is chosen empirically, so that the proposal density closely approximates the state space distribution around the initial state $\boldsymbol{x}_k(1)$.

The distribution π for evaluating the samples in (15) can be obtained by Bayes' rule, assuming that $\boldsymbol{\theta}_k$ is independent of $\boldsymbol{x}_k(t)$, as follows:

$$\pi = \frac{p(\boldsymbol{z}(1)|\boldsymbol{x}_k(1), \boldsymbol{\theta}_k))p(\boldsymbol{x}_k(1)|\boldsymbol{\theta}_k)}{p(\boldsymbol{z}(1)|\boldsymbol{\theta}_k)} \tag{17}$$

where the likelihood distribution $p(\boldsymbol{z}(1)|\boldsymbol{x}_k(1), \boldsymbol{\theta}_k)$ is obtained from the measurement process (2) and the prior distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{\theta}_k) = p(\boldsymbol{x}_k(1))$ is assumed to be uniform. The value of $p(\boldsymbol{z}(1)|\boldsymbol{\theta}_k)$ is irrelevant for the purposes at hand, since it is independent of $\boldsymbol{x}$ and hence cancels in (15).

B. The M-Step

In order to illustrate the flexibility of the proposed method, we demonstrate the M–step for the general case where no structure is assumed on $\boldsymbol{h}(\cdot)$. Here, $\boldsymbol{h}(\cdot)$ is modelled as a mixture of Gaussian kernels. The case where the model $\boldsymbol{h}(\cdot)$ is known up to unknown parameters $\boldsymbol{\theta}$ is considered later.

By substituting the posterior distribution of the states given the observations obtained in the E-step ($U_{k+1}^\star = p(\boldsymbol{x}_k|\boldsymbol{z}, \boldsymbol{\theta}_k)$) into (5), the required optimization for the M-step of the kth EM iteration becomes

$$\boldsymbol{\theta}_{k+1} = \arg\max_{\boldsymbol{\theta}} \int_\chi p(\boldsymbol{x}_k|\boldsymbol{z}, \boldsymbol{\theta}_k) \log p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{x}_k(1) \ldots d\boldsymbol{x}_k(L). \tag{18}$$

To proceed with this optimization, we incorporate our model for the observation function $\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{\theta})$ in (2). This function is modelled as a mixture of Gaussians with P components, as

$$h\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big) \approx \sum_{p=1}^{P} \boldsymbol{m}_p g_p\big(\boldsymbol{x}(t)\big) + \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b} \tag{19}$$

where the parameters $\boldsymbol{m}_p \in \mathbb{C}^J$ are the coefficients of the scalar Gaussian kernels g_p, with fixed centers $\boldsymbol{c}_p \in \mathbb{C}^M$ and fixed covariance matrices $\boldsymbol{S}_p \in \mathbb{C}^{M\times M}$. The centers are distributed uniformly over the range of $\boldsymbol{x}$, and the covariances $\boldsymbol{S}_p$ may be assigned arbitrarily.[2] The Gaussian kernels are defined as:

$$g_p(\boldsymbol{x}) = (2\pi)^{M/2}|\boldsymbol{S}_p|^{-\frac{1}{2}} \exp[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{c}_p)^T \boldsymbol{S}_p^{-1}(\boldsymbol{x} - \boldsymbol{c}_p)]. \tag{20}$$

The quantity $\boldsymbol{A} \in \mathbb{C}^{J\times M}$ is a constant matrix, $\boldsymbol{b} \in \mathbb{C}^J$ is a constant bias term. The matrix $\boldsymbol{\theta} \in \mathbb{C}^{J\times(P+M+1)}$ is therefore defined as

$$\boldsymbol{\theta} \triangleq [\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_P, \boldsymbol{A}, \boldsymbol{b}], \tag{21}$$

which according to the assumptions, is time–invariant. We also define the vector $\boldsymbol{\Phi}(t) \in \mathbb{C}^{(P+M+1)\times 1}$, which includes the time–varying parameters in $\boldsymbol{h}(\cdot)$ as

$$\boldsymbol{\Phi}(t) \triangleq [g_1(\boldsymbol{x}(t)), g_2(\boldsymbol{x}(t)), ..., g_P(\boldsymbol{x}(t)), \boldsymbol{x}(t)^T, 1]^T. \tag{22}$$

Then, (19) can be written in the form

$$\boldsymbol{h}\big(\boldsymbol{x}(t), \boldsymbol{\theta}\big) \approx \boldsymbol{\theta}\boldsymbol{\Phi}(t). \tag{23}$$

If some structure on $\boldsymbol{h}(\cdot)$ is available, e.g., the model is known up to some unknown parameters $\boldsymbol{\theta}$, then this structure should be used instead of the form given by (23).

We now evaluate the probability distribution $p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}_k)$ at EM iteration k in (18). This may be evaluated according to

$$\begin{aligned} p(\boldsymbol{x}_k, \boldsymbol{z}|\boldsymbol{\theta}_k) &= p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)\ p(\boldsymbol{x}_k|\boldsymbol{\theta}_k) \\ &\propto p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k) \end{aligned} \tag{24}$$

where the second line follows because the prior distribution of the states is assumed to be independent of the unknown parameters, and is assigned a uniform distribution. The distribution $p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)$ is easily obtained as the likelihood distribution obtained using the observation equation (2).

The log-likelihood $\log p(\boldsymbol{z}|\boldsymbol{x}_k, \boldsymbol{\theta}_k)$ of a single fully observed data point $\boldsymbol{z}(t)$ under the model at EM iteration k, using (23) and (24) is then given as

$$-\big[\boldsymbol{z}(t) - \boldsymbol{\theta}_k\boldsymbol{\Phi}_k(t)\big]^H \boldsymbol{Q}_k^{-1}\big[\boldsymbol{z}(t) - \boldsymbol{\theta}_k\boldsymbol{\Phi}_k(t)\big] - \ln|\boldsymbol{Q}| + \text{constant}, \tag{25}$$

[2]In the following simulations, they were all assigned to the identity matrix.

where $\boldsymbol{Q}_k$ is the estimate of the observation noise covariance defined in (2), at EM iteration k. By substituting the model log-likelihood into (18), and combining the terms for $t = 1, \ldots, L$, the relevant M-step optimization is then

$$\min_{\boldsymbol{\theta},\boldsymbol{Q}} \left\{ \int_{\chi} \sum_{t=1}^{L} p(\boldsymbol{x}_k(t)|\boldsymbol{z},\boldsymbol{\theta}_k) \big[\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)\big]^H \boldsymbol{Q}^{-1} \big[\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)\big] d\boldsymbol{x} + \ln|\boldsymbol{Q}| \right\}. \tag{26}$$

By denoting the expectation over the posterior distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z},\boldsymbol{\theta}_k)$ by $\big\langle\big\rangle$, the objective function then becomes

$$\min_{\boldsymbol{\theta},\boldsymbol{Q}} \sum_{t=1}^{L} \left\langle [\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1} [\boldsymbol{z}(t) - \boldsymbol{\theta}\boldsymbol{\Phi}_k(t)] \right\rangle + \ln|\boldsymbol{Q}_k|. \tag{27}$$

It is shown in the Appendix that the solution to the above is given by

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k+1} = \left\langle \sum_{t=1}^{L} \boldsymbol{z}(t)\boldsymbol{\Phi}_k^H(t) \right\rangle \left\langle \sum_{t=1}^{L} \boldsymbol{\Phi}_k(t)\boldsymbol{\Phi}_k(t)^H \right\rangle^{-1} \tag{28}$$

$$\hat{\boldsymbol{Q}} = \boldsymbol{Q}_{k+1} = \sum_{t=1}^{L} \left\langle \boldsymbol{z}(t)\boldsymbol{z}^H(t) \right\rangle - \sum_{t=1}^{L} \left\langle \hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)\boldsymbol{z}^H(t) \right\rangle, \tag{29}$$

where $L \geq J$. Thus, given the expectations in the angular brackets, the optimal parameters can be obtained by solving a set of linear equations.

The expectations above are evaluated using the particle filter. Given the computed particles $\boldsymbol{x}_k^{(i)}(t)$ and using the approximation (8) for the posterior distribution, the expectation of any function $f(\boldsymbol{x})$ can be approximated in a manner similar to that of (14) by

$$\langle f(\boldsymbol{x})\rangle = \int_{\chi} f(\boldsymbol{x}_k(t)) p(\boldsymbol{x}_k(t)|\boldsymbol{z},\boldsymbol{\theta}_k) d\boldsymbol{x}_k(t) \approx \sum_{i=1}^{N} w_k^{(i)}(t|L) f(\boldsymbol{x}_k^{(i)}(t)). \tag{30}$$

The number P of Gaussian kernels is chosen empirically, so that the kernels are positioned sufficiently densely over the region of support of the state variables.

Initialization of the EM Algorithm: The initial parameters $\boldsymbol{\theta}_1$ must be chosen with some care, otherwise the EM algorithm may not converge. In the experiments described in the following sections, successful results were obtained by assigning the Gaussian kernel coefficients $\boldsymbol{m}_p$ to equal values, the kernel centers $\boldsymbol{c}_p$ to a uniformly–spaced grid, and the kernel covariance matrices $\boldsymbol{S}_p$ to the identity. (The parameters $\boldsymbol{c}_p$ and $\boldsymbol{S}_p$ are held fixed throughout the EM iterations.) The matrix $\boldsymbol{A}$ is also assigned to be an identity (padded appropriately with zeros), and the bias $\boldsymbol{b}$ to zero. Given this initial $\boldsymbol{\theta}$, an E–step was performed to obtain the initial states, $\boldsymbol{x}_0(t), t = 1, \ldots, L$.

C. Summary

Here we give a step–by–step overview of the proposed EM–PF algorithm:

- **Initialization:** ($k = 1$) Given a set of measurements, $\boldsymbol{z} = \{\boldsymbol{z}(t), t = 1, \ldots, L\}$, initialize the parameter vector $\boldsymbol{\theta}_1$ to suitable values, as described immediately above.
- **Iterate the EM Algorithm: for** $k = 1, 2, \ldots$
  - **E–step:** In the E-step, we estimate the states $\boldsymbol{x}_k(t)$ using the particle filtering approximation $\hat{p}_N(\boldsymbol{x}_k(t)\big|\boldsymbol{z}, \boldsymbol{\theta}_k)$ to the posterior distribution $p(\boldsymbol{x}_k(t)\big|\boldsymbol{z}, \boldsymbol{\theta}_k)$ with the most current model. More detail is given as follows:
    - Initialize the posterior distribution $p(\boldsymbol{x}_k(1)|\boldsymbol{z}(1), \boldsymbol{\theta}_k)$ at the current EM iteration k using the Metropolis–Hastings MCMC method, described in Sect. 3-A. The required proposal density $q(\cdot|\cdot)$ for the MH-MCMC algorithm is chosen to be a Gaussian distribution with mean equal to the previous state $\boldsymbol{x}^{(i-1)}$ and variance chosen so that the support of the function covers adequate space around the current state. Set the filtering weights $w_k^{(i)}(t) = 1, i = 1, \ldots, N$.
    - propagate $p(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ for $t = 2, 3, \ldots, L$ using the particle filter. The approximate posterior distribution $\hat{p}_N(\boldsymbol{x}_k(t)|\boldsymbol{z}, \boldsymbol{\theta}_k)$ for the fixed–interval smoothing case is given by (12) as a function of the smoothing weights $w_k^{(i)}(t|L)$. These weights are propagated to the next time step by the following procedure: first the *filtering* weights $w^{(i)}(t-1)$ are propagated to time t using (9) and (10). Then, the filtering weights are converted to the smoothing weights $w_k^{(i)}(t|L)$ using the algorithm surrounding (13).
    - The likelihood $p(\boldsymbol{z}(t)|\boldsymbol{x}_k(t), \boldsymbol{\theta}_k)$ used in (9) is given from the MoG model (19), knowing the statistics of $\boldsymbol{n}$. The form of prior distribution $p(\boldsymbol{x}(t+1)|\boldsymbol{x}(t))$ also used in (9) depends on the underlying physics of the model, as determined by (2). Examples are given in Sects. 4 and 5.
    - Once the smoothing weights $w_k^{(i)}(t|L)$ are available, an approximate conditional mean estimate of the states $\boldsymbol{x}_k(t)$ is given at each time through (14).
  - **M–step**: The approximated states with their corresponding measurements are then used in the M-step to re-estimate the parameters of the MoG, i.e., the parameter vector $\boldsymbol{\theta}_{k+1}$ and the model noise covariance $\boldsymbol{Q}_{k+1}$ using (28) and (29) respectively. These estimated parameters are used in the next E-step. The necessary expectations in these two equations are evaluated using (30).

In the following two sections, we apply the proposed EM–PF method to solve the *bearing-only tracking* problem with uncertain model parameters, and the so-called *sensor registration* problem in a multi-sensor scenario.

4. BEARING-ONLY TARGET TRACKING WITH BIASED MEASUREMENTS

A. Problem Statement

We apply the EM-PF algorithm to a bearing-only target tracking problem in the presence of sensor bias. Even though known structure which may be exploited does exist in the observation model $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta})$ in this case. However in this treatment we choose to ignore it, and use the model given by (23) instead. This is done to demonstrate that useful state information can be estimated with limited knowledge of the model.

The problem consists of a linear state transition and a nonlinear measurement process. The problem is defined in [1]. In this scenario, a platform with a sensor moves according to the discrete time equations:

$$x_p(t) = \overline{x}_p(t) + \Delta x_p(t), \quad y_p(t) = \overline{y}_p(t) + \Delta y_p(t), \qquad t = 1, 2, \ldots \tag{31}$$

where $\overline{x}_p(t)$ and $\overline{y}_p(t)$ are the average platform position coordinates, and the perturbations $\Delta x_p(t)$ and $\Delta y_p(t)$ are assumed to be mutually independent zero-mean Gaussian white noise sequences with variances r_x and r_y, respectively. The average (unperturbed) platform motion is assumed to be horizontal with constant velocity. Its position as a function of the discrete time t (in meters) is:

$$\overline{x}_p(t) = a_1 t, \qquad \overline{y}_p(t) = a_2 \tag{32}$$

where a_1 and a_2 are constants.

It is assumed a target moves on the $\boldsymbol{x}$-axis according to

$$\boldsymbol{x}(t+1) = \boldsymbol{F}(t)\boldsymbol{x}(t) + \boldsymbol{w}(t) \tag{33}$$

where:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \qquad \boldsymbol{F}(t) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{34}$$

and x_1 and x_2 denote the position and velocity of the target, $T = 1s$ is the normalized sampling period, and $\boldsymbol{w}(t) \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$, where

$$\boldsymbol{\Sigma}_{\boldsymbol{w}} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \tag{35}$$

and q is a scalar. The sensor measurement process is:

$$z(t) = h(\boldsymbol{x}_p(t), \boldsymbol{y}_p(t), x_1(t)) + \beta + v_s(t), \tag{36}$$

where

$$h(\boldsymbol{x}_p(t), \boldsymbol{y}_p(t), x_1(t)) = \tan^{-1} \frac{y_{1p}(t)}{x_1(t) - x_{1p}(t)} \tag{37}$$

is the bearing between the horizontal and the line of sight from the sensor to the target, and the sensor noise $v_s(t)$ is zero mean white Gaussian with variance r_s. The sensor noise is assumed to be independent of the sensor platform perturbations. Also β is the unknown bias of the measurements.

The estimation of the target's state is performed using *only* the measurements (36).

The platform location perturbations induce additional errors in the measurements. The effect of these errors is evaluated by expanding the nonlinear measurement function h in a Taylor series about the average platform position. The resulting measurement process can then be written as:

$$z(t) = h(\overline{\boldsymbol{x}}_p(t), \overline{\boldsymbol{y}}_p(t), x_1(t)) + v(t) = \tan^{-1} \frac{\overline{y}_{1p}(t)}{x_1(t) - \overline{x}_{1p}(t)} + \beta + v(t), \tag{38}$$

where the equivalent measurement noise $v(t)$ is zero mean white Gaussian with variance given by

$$E[v(t)^2] \triangleq r(t) = \frac{(\overline{y}_{1p}(t))^2 r_x + (x_1(t) - \overline{x}_{1p}(t))^2 r_y}{\{(\overline{y}_{1p}(t))^2 + (x_1(t) - \overline{x}_{1p}(t))^2\}^2} + r_s(t). \tag{39}$$

Notice that the variance of the equivalent measurement noise is time varying. For more details on modelling the new measurement process refer to [1].

In the following, we use the EM-PF algorithm to track the target corresponding to the uncertain observation model which has been discussed. This method ignores any known structure in the model. No doubt better performance could be achieved if a method which exploits the model structure of (38) were used, where $\overline{y}_{1p}(t), \overline{x}_{1p}(t)$ and β were treated as unknown parameters. Despite this fact, this example successfully demonstrates that the EM-PF method can be applied to the problem of bearing–only tracking with model uncertainty in the form of sensor bias. The example also demonstrates that the EM-PF method can be successfully applied to a range of problems where little is known about the structure of the observation model.

B. Simulation Results

In this simulation scenario, the parameter values are listed in Table 4.1. The measurements are biased by a value of $\beta = 0.5$ radians (see Figure 2).

It is important to compare the performance of the optimal smoother and that of the proposed EM–PF algorithm. In general, the optimal smoother is analytically intractable or prohibitively expensive to run. However, even if the optimal smoother is impossible to run, is still possible to determine the performance of the optimal smoother in terms of some performance criterion such as the mean–squared error (MSE). Indeed, if we resort to the computation of the posterior Cramér-Rao lower bound (PCRLB) as described in [29], [30], we can determine the achievable MSE of the generally intractable

TABLE 4.1

PARAMETERS FOR THE BEARING–ONLY TRACKING SIMULATION EXAMPLE.

Parameter symbol	Meaning	Value
r_x	variance of $\Delta \boldsymbol{x}_p$ in (31)	$1m^2$
r_y	variance of $\Delta \boldsymbol{y}_p$ in (31)	$1m^2$
a_1	see (32)	4 m/sec
a_2	see (32)	20 m
q	covariance scalar in (35)	0.01 $\mathrm{m}^2/\mathrm{sec}^3$
r_s	measurement noise variance in (38)	$5.24 \times 10^{-3}\mathrm{rad}^2$
$\boldsymbol{x}_0$	initial condition for the state	$[80, 1]^T$
P	Number of Gaussian kernels in the MoG model	20
L	Number of observations	21
N	Number of particles	200
J	Number of sensors	1
M	Number of state variables	2

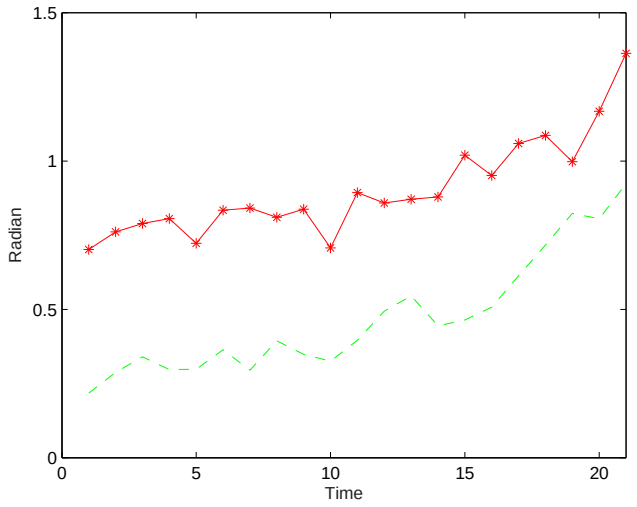


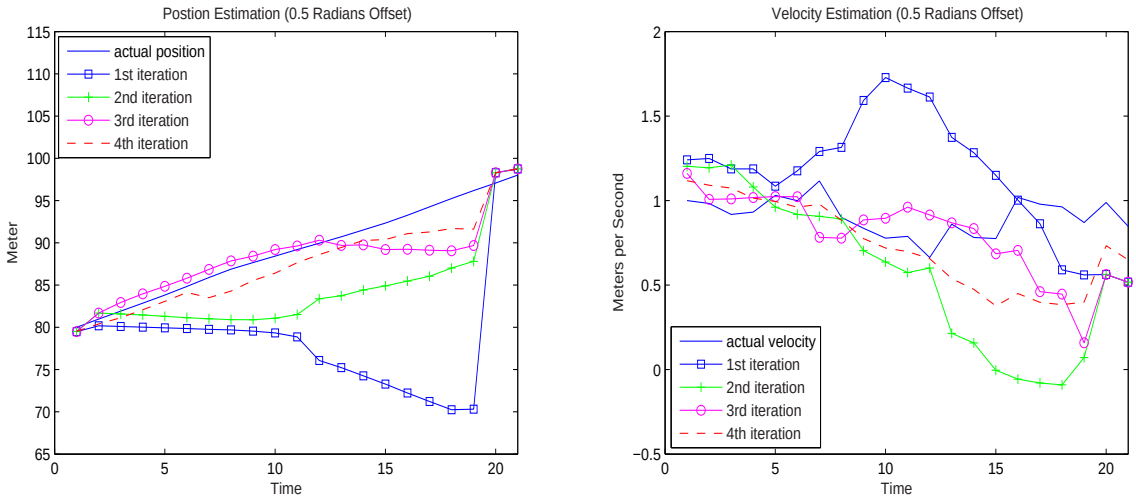
Fig. 2. Unbiased, noise-free but perturbed measurements obtained from (36) with $\beta = 0$ (bottom), and the biased, perturbed, noisy measurements from (38), that are input to the algorithm (top).

optimal smoother. More importantly, we can obtain a theoretical benchmark for any other practical suboptimal smoothing algorithm.

Formally, the PCRLB for *fixed–interval smoothing* can be stated as follows:

$$\boldsymbol{M}_k = E\left[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k(t))(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k(t))^T\right] \geq \boldsymbol{J}_k^{-1} \tag{40}$$

where $\boldsymbol{M}_k$ is the MSE correlation matrix and $\boldsymbol{J}_k^{-1}$ denotes a matrix which can be recursively computed as described in [29]. We stess that $\hat{\boldsymbol{x}}_k(t)$ need not be an unbiased estimator, and that (40) is a matrix



(a) (b)

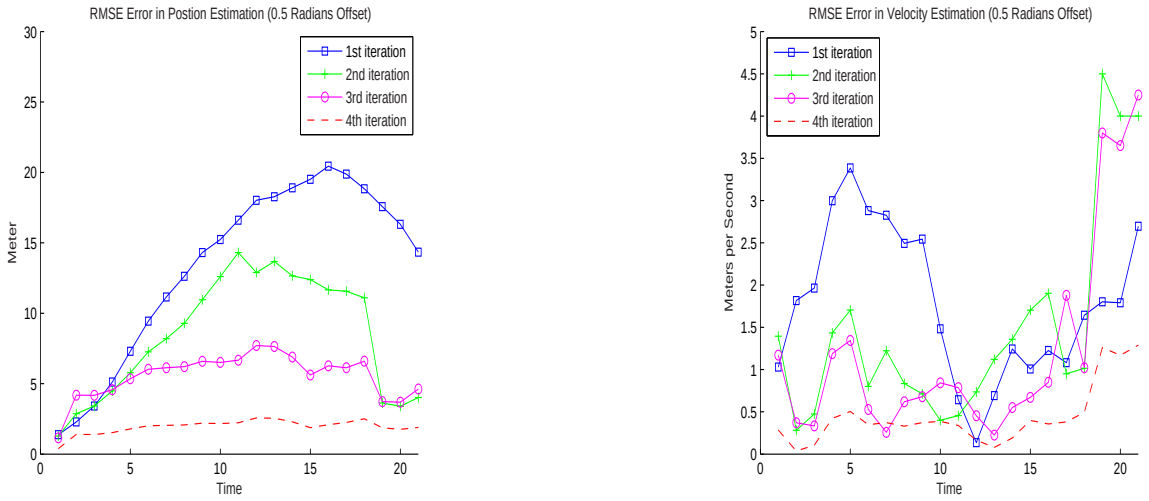
Fig. 3. Position (a) and velocity (b) tracking trajectories for the EM-PF algorithm over four successive iterations.

inequality in the sense that $\boldsymbol{M}_k - \boldsymbol{J}_k^{-1}$ is a positive semi-definite matrix. In general, (40) provides a lower bound on the MSE of the considered estimator $\hat{\boldsymbol{x}}_k(t)$.

Figure 3 shows the position and the velocity tracking trajectories, respectively, over four successive iterations of the EM-PF algorithm for the bearing-only tracking problem for a typical run. Also, Figure 4 shows the root MSE error for tracking the position and velocity of the target over 50 Monte-Carlo runs, respectively. The figures represent the error for four consecutive iterations of the algorithm.

Figure 5 shows the position and velocity root MSE's of the EM-PF algorithm at the fourth iteration for the same run. Also shown are the corresponding PCRLB curves. In this case, the PCRLB assumes the model is known, except that the biases are unknown random variables, constant over the observation interval. It is noted that the performance of the EM-PF root MSE is worse than the PCRLB. This discrepancy is to be expected, since the PCRLB results correspond to a known observation model, (except for the bias), whereas the EM-PF algorithm assumes no knowledge of the model. Also, there are errors in the MoG and particle filter approximations. As can be seen, the EM-PF algorithm is capable of managing uncertain dynamics by identifying them, and then using this information for better estimation of the states. Although the simulation results are provided for Gaussian nonlinear measurement dynamics, the EM–PF algorithm is nevertheless capable of handling the non–Gaussian case.

We also present results in Figure 6 showing root MSE vs. observation noise variance (q in (35)) averaged over 50 simulation runs. We can see that the root MSE increases relatively smoothly with



(a) (b)

Fig. 4. Root MSE of the position (a) and velocity (b) state estimates vs. time over 50 Monte–Carlo runs of the EM-PF algorithm for four iterations.

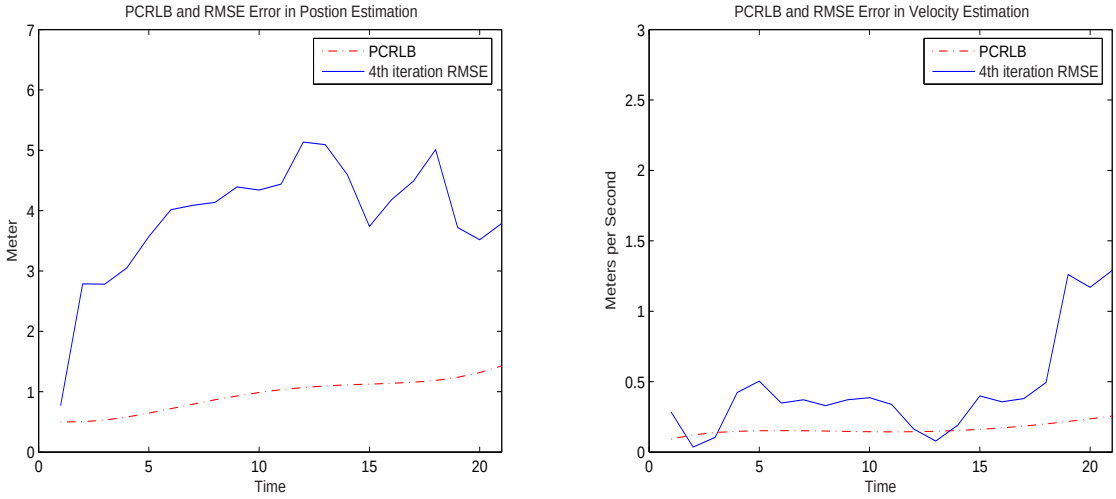
increasing noise variance until a threshold is reached at a noise variance value of about 10^{-2}, beyond which the method breaks down. The root MSE does not steadily decrease to zero with decreasing noise variance, due to the errors in the MoG model and the error in the particle filter approximations.

We now compare the performance obtained from the EM-PF algorithm with the interactive multiple model (IMM) approach [1] for the same bearing–only tracking problem, where each model uses an extended Kalman filter (EKF) with a different range of bias[3]. In the following experiments, we used 16 models, whose corresponding bias values are uniformly distributed over 0 to 1 radian. The various parameters describing the problem were the same as those for the EM–PF case, and are given in Table 4.1.

Figure 7 shows the position and velocity estimation performance for the IMM algorithm. The corresponding RMSE's are shown in Figure 8 over the time interval $[0, 40]$s. The RMSE's for the interval $[20, 40]$s are shown in Figure 9.

It may be observed that, over the interval $[20, 40]$s after which the IMM method has acquired track, the performance of the EM–PF and IMM algorithms are roughly equivalent. However, it may be observed that the IMM model requires about 20 time steps to acquire track, because of the time required to assess the individual model probabilities and determine the winner. However, the EM-PF approach, due to its MCMC initialization procedure, requires virtually no time for acquisition. Further, any approach

[3]Experiments were also conducted using a conventional EKF, using a model which did not incorporate bias. In this case, the track loss rate approached 100%.



(a) (b)

Fig. 5. Root MSE of the position (a) and velocity (b) state estimates vs. time of the EM-PF algorithm at the fourth iteration, along with the corresponding PCRLB curves.

using EKFs must have available an accurate observation model, which is not required for the EM-PF method.

It is straightforward to modify the proposed EM–PF algorithm so that it can handle time variations in the model parameters $\boldsymbol{\theta}$. In this vein, a hypothetical experiment was conducted where the bias changed sign half–way through the observation record. It was observed that the EM-PF method quickly adapted to this change in model parameters.

5. Sensor Registration

We first introduce the fundamental idea of the sensor registration problem. An example of this problem is in the tracking scenario where multiple targets are being tracked by multiple sensors. The locations of the sensors are determined by a Cartesian coordinate system, while measurements from the sensors are obtained in polar coordinates. To properly combine the measurements in a multisensor scenario, it is required to transform the measurements into a common reference frame free from sensor registration errors. In a multi–sensor scenario, sensor registration errors can cause significant error in the target location. Biased measurements, for example, can increase estimation error or even corrupt the estimation process completely.

Bias estimation is inevitable in current multisensor estimation scenarios. The classical approach to mitigate this problem is to firstly transform the measurements into a common coordinate system,

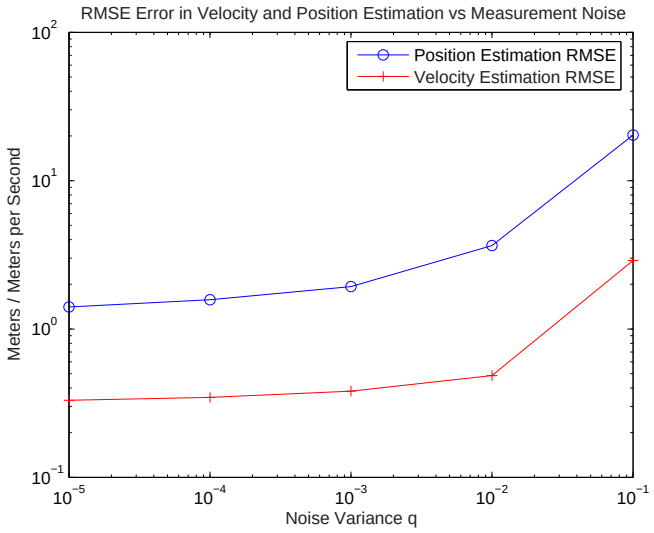
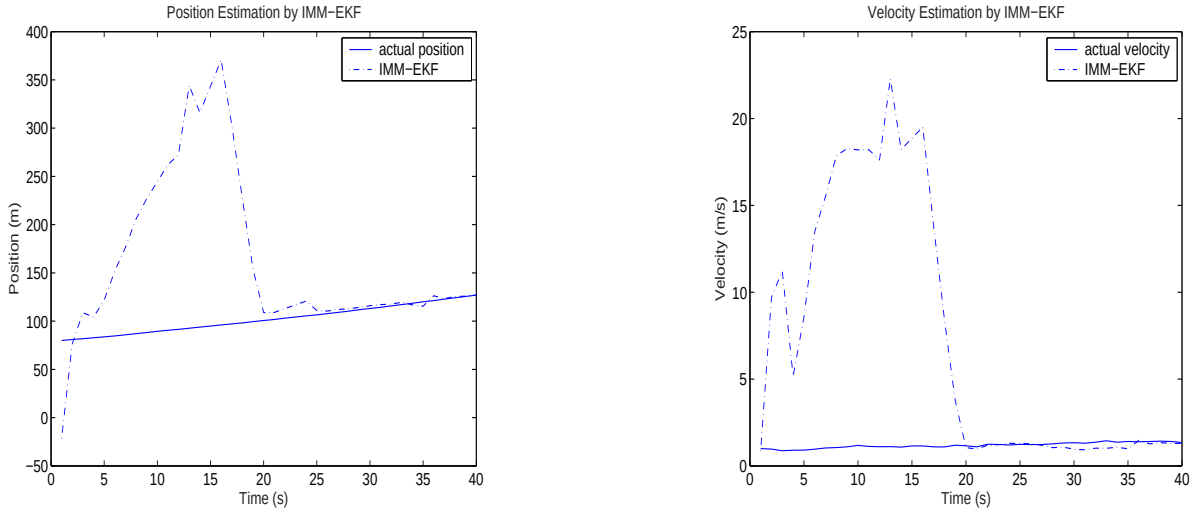


Fig. 6. Root MSE vs. observation noise variance for the bearing–only tracking problem.

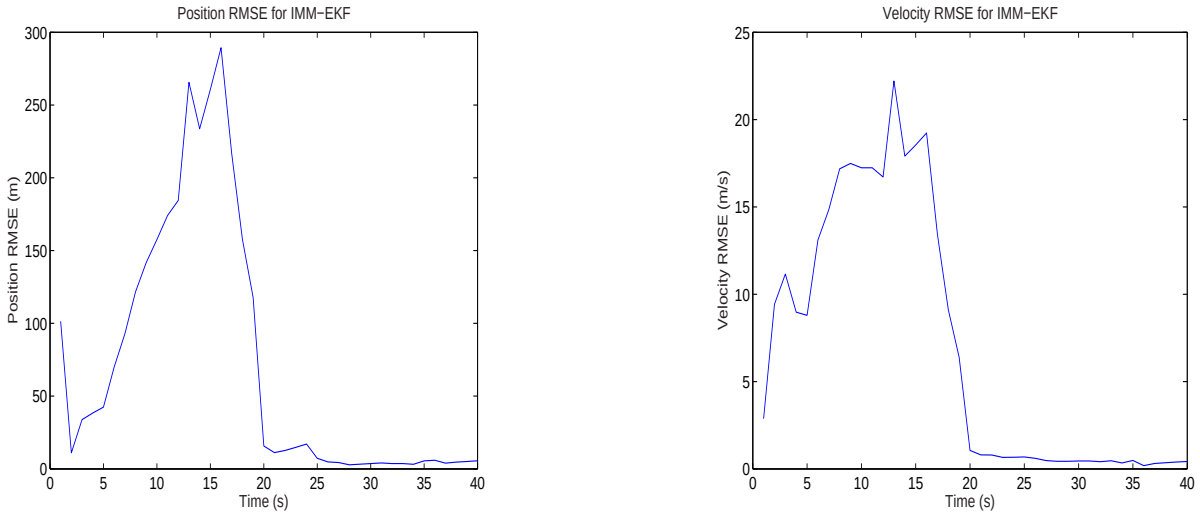


(a) (b)

Fig. 7. Position and velocity estimates for the IMM-EKF algorithm, applied to the bearing-only tracking problem.

estimate the biases by a batch algorithm and then remove the bias from the subsequent measurements. The EM-PF algorithm can be applied in this regard. The EM-PF algorithm may be considered similar to the recently reported method called maximum likelihood registration (MLR) [31] that indirectly estimated sensor biases and removes the effect of them in the estimation process.

In surveillance applications, it is known that the *stereographic projection* of three dimensional data onto a two-dimensional plane introduces error in sensor registration [31]. We overcome this problem using geodetic transformations for mapping the sensor measurements into the earth centered earth fixed



(a) (b)

Fig. 8. Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem over the time interval 0 to 40s.

(ECEF) coordinate system. Sensor registration is then performed in the ECEF coordinate system. The performance of the EM-PF algorithm is determined using simulations based on a scenario presented previously in [31].

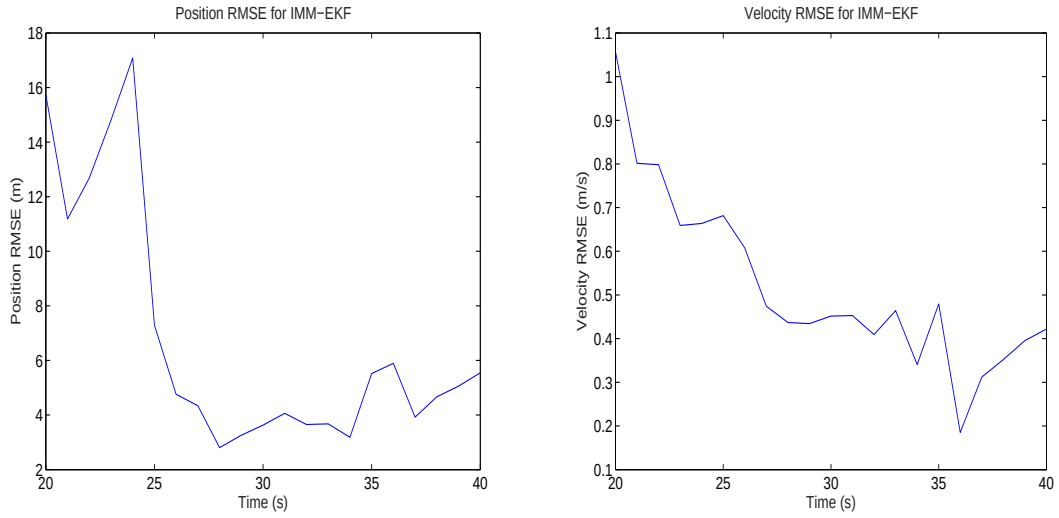
A. Problem Statement

The problem definition provided in this section is based on the presentation in [31]. The state vector $\boldsymbol{x}(t)$ of a moving target at time t consists of the three-dimensional position of the target defined in ECEF coordinate system:

$$\boldsymbol{x}(t) = [X(t) \quad Y(t) \quad Z(t)]^T. \tag{41}$$

The origin of the ECEF coordinate system is at the center of the Earth. The X axis is extends from origin to the intersection of the prime meridian (0^o longitude) and the equator (0^o latitude). In the right-handed coordinate system, the Y axis extends from the origin to the intersection of the 90^o longitude and the equator. Also the Z axis passes through the origin and the north pole (90^o latitude).

Consider M sensors located at $(L_m, \lambda_m, \alpha_m)$ $(m = 1, ..., M)$, where L_m is the geodetic latitude, λ_m is the longitude and α_m is the altitude above the reference ellipsoid, in the *geodetic* coordinate system. At time instant t, the mth sensor measures the position of a common target in terms of a



(a) (b)

Fig. 9. Position and velocity estimate RMSEs for the IMM-EKF algorithm, applied to the bearing-only tracking problem, for the time interval from 20s to 40s.

three-dimensional measurement vector $\boldsymbol{z}_m(t)$[4]:

$$\boldsymbol{z}_m(t) = [\rho_m(t) \quad \gamma_m(t) \quad \epsilon_m(t)]^T, \qquad m = 1, \ldots, M, \tag{42}$$

where M is the number of sensors, ρ_m is slant range, γ_m is azimuth (measured clockwise from North), and ϵ_m is elevation, each with respect to the mth sensor. The registration vector for each sensor also consists of the corresponding biases, i.e.,

$$\boldsymbol{\beta}_m = [\Delta\rho_m \ \Delta\gamma_m \ \Delta\epsilon_m]^T. \tag{43}$$

In order to register the sensor measurements in a common coordinate system, we transform the sensor position data into the ECEF coordinate system. Given the sensor position $(L_m, \lambda_m, \alpha_m)$ the following equations give the ECEF coordinates (X_m, Y_m, Z_m):

$$X_m = (c + \alpha_m).\cos L_m.\cos\lambda_m \tag{44}$$

$$Y_m = (c + \alpha_m).\cos L_m.\sin\lambda_m \tag{45}$$

$$Z_m = (c(1 - e^2) + \alpha_m).\sin L_m, \tag{46}$$

[4]Note that the adopted notation implies that non–time varying coordinates specify *sensor* locations, whereas time varying coordinates specify *target* locations.

in which we adopt an $WGS-84$ ellipsoid[5] with parameters:

$$c = a/\sqrt{1-e^2\ \sin^2 L_m} \tag{47}$$

$$e = 0.0818 \tag{48}$$

$$a = 6378137.0\ \text{m}. \tag{49}$$

Since the state vector consisting of the position of the target (as well as the position of sensors) is defined in ECEF coordinates, and the measurements are in polar coordinates, it is difficult to write the explicit dependence of the measurement functions h on the state vector. Instead, we proceed to transform the state vectors into polar coordinates and model the measurement process in this system. Define the target state vector in the local tangent plane of sensor m as:

$$\boldsymbol{x}_m(t) = [\varepsilon_m(t)\ \ \nu_m(t)\ \ \upsilon_m(t)]^T, \tag{50}$$

where $\varepsilon_m(t)$, $\nu_m(t)$ and $\upsilon_m(t)$ denote east, north, and up axes at sensor m. These components are computed in terms of the state vector given by (41) and the position of the sensors as follows:

$$\varepsilon_m(t) = -(X(t)-X_m)\sin\lambda_m + (Y(t)-Y_m)\cos\lambda_m, \tag{51}$$

$$\nu_m(t) = -(X(t)-X_m)\sin L_m\cos\lambda_m - (Y(t)-Y_m)\sin L_m\sin\lambda_m + (Z(t)-Z_m)\cos L_m \tag{52}$$

$$\upsilon_m(t) = (X(t)-X_m)\cos L_m\cos\lambda_m + (Y(t)-Y_m)\cos L_m\sin\lambda_m + (Z(t)-Z_m)\sin L_m. \tag{53}$$

Now we can express the nonlinear measurement functions ($h_{mj};\ \ m=1,...,M;\ j=1,2,3$ for sensor m and the three measurement components, in terms of the state vector (50) as follows:

$$h_{m1}\left(\boldsymbol{x}_m(t)\right) = \sqrt{\varepsilon_m^2(t)+\nu_m^2(t)+\upsilon_m^2(t)}, \tag{54}$$

$$h_{m2}\left(\boldsymbol{x}_m(t)\right) = \tan^{-1}\left(\frac{\varepsilon_m(t)}{\nu_m(t)}\right), \tag{55}$$

$$h_{m3}\left(\boldsymbol{x}_m(t)\right) = \sin^{-1}\left\{\frac{\upsilon_m(t)}{\sqrt{\varepsilon_m^2(t)+\nu_m^2(t)+\upsilon_m^2(t)}}\right\}. \tag{56}$$

Having prepared the necessary definitions and assuming that the location of the static sensors are known perfectly, we can now define the measurement process for sensor m as:

$$\boldsymbol{z}_m(t) = \boldsymbol{h}_m\left(\boldsymbol{x}_m(t)\right) + \boldsymbol{\beta}_m + \boldsymbol{v}_m(t), \qquad m=1,\ldots,M, \tag{57}$$

where $\boldsymbol{z}_m(t) \in \Re^{3\times 1}$ consists of the three measurement components range (ρ), azimuth (γ), and elevation (ϵ) of the target, respectively. Also, $v_m \in \Re^{3\times 1}$ is the random measurement noise vector assumed to be *i.i.d.* white noise, mutually independent from component to component, with covariance matrix $\boldsymbol{\Sigma}_{z_m} = \text{diag}(\sigma^2_{\rho_m}, \sigma^2_{\gamma_m}, \sigma^2_{\epsilon_m})$. The nonlinear functions $h_{mj},\ j=1,2,3$, are defined in (54) –

[5]The world geodetic system (1984) (WGS-84) is a standard for earth coordinate systems. The WGS-84 ellipsoid minimizes the error between itself and the true shape of earth over a specific region of interest.

(56). For M sensors measuring the location of the common target, the measurements from (57) can be combined into a single equation as follows:

$$\boldsymbol{z}(t) = \boldsymbol{h}\left(\boldsymbol{x}(t)\right) + \boldsymbol{\beta} + \boldsymbol{v}(t), \tag{58}$$

where $\boldsymbol{v}(t) = [v_1(t)^T ... \ v_M(t)^T]^T$ is the random measurement noise vector assumed to be an *i.i.d.* white noise process, mutually independent from sensor to sensor. The measurement covariance matrix is $\boldsymbol{\Sigma}_z = \text{diag}(\Sigma_{z_1}, \ ..., \ \Sigma_{z_M}) \in \Re^{3M \times 3M}$. The vectors $\boldsymbol{x}(t) = [\boldsymbol{x}_1(t)^T ... \ \boldsymbol{x}_M(t)^T]^T$ and $\boldsymbol{z}(t) = [\boldsymbol{z}_1(t)^T ... \ \boldsymbol{z}_M(t)^T]^T$ are the state variable and the noisy output measurement vectors for the M sensors, respectively. Also, the vector valued nonlinear function $\boldsymbol{h}_{mj}, m = 1, ..., M; j = 1, 2, 3$ is assumed to be known for the M sensors and the three values of the measurement vectors $\boldsymbol{z}_m(t) = [\rho_m(t) \ \ \gamma_m(t) \ \ \epsilon_m(t)]^T$. The vector $\boldsymbol{\beta} = [\beta_1^T ... \ \beta_M^T]^T \in \Re^{3M}$ contains the unknown biases for the M sensors that is assumed to be deterministic, time-invariant and independent of the state vector $\boldsymbol{x}(t)$.

The state process is assumed to be modelled by a linear first-order Markov process as follows:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \boldsymbol{w}(t), \tag{59}$$

where $\boldsymbol{w}(t) \in \Re^3$ is an *i.i.d.* noise process with covariance matrix $\boldsymbol{R} = \text{diag}(\sigma_x, \ \sigma_y, \ \sigma_z)$, and $\boldsymbol{x}(t) = [X(t) \ \ Y(t) \ \ Z(t)]^T$ is position of the target in the ECEF coordinate system.

Given L measurement vectors $\boldsymbol{z}(t), \ t = 1, \ldots, L$, the problem is to remove the effect of the measurement biases $\boldsymbol{\beta}$ and to estimate the states $\boldsymbol{x}(t), \ t = 1, \ldots, L$ accurately.

B. Simulation Results

We implement exactly the same scenario in [31] to compare the performance of the EM-PF algorithm with the recently reported MLR algorithm for sensor registration example. The details of the simulation scenario are given here from the stated reference:

There are two ground-based sensors measuring the position of a moving target. The geodetic coordinate of sensors, $(L_m, \lambda_m, \alpha_m)$, are: $(-12^o 30', 131^o 6', 15\text{m})$ for sensor 1 and $(-14^o 18', 129^o 36', 10\text{m})$ for sensor 2. The target is flying from geodetic coordinate $(-12^o, 129^o 30', 10\text{km})$ to $(-13^o 30', 130^o 30', 10\text{km})$, then it makes a mild turn and finished at $(-14^o, 131^o 12', 10\text{km})$. A total of $K = 120$ synchronous pairs of measurements are collected. We assign $\sigma_x = \sigma_y = \sigma_z = 10^2 \ \text{m}^2/\text{s}^4$).

The true sensor biases used in simulations are as follows. Sensor 1: $\Delta\rho_1 = 2.5 \ \text{km}$; $\Delta\gamma_1 = -2.5^o$; $\Delta\epsilon_1 = -0.5^o$. Sensor 2: $\Delta\rho_2 = -1.8 \ \text{km}$; $\Delta\gamma_2 = 3^o$; $\Delta\epsilon_2 = 1^o$. Measurement noise is zero-mean Gaussian with covariance $\Sigma_{z_m} = \text{diag}(\sigma^2_{\rho_m}, \sigma^2_{\gamma_m}, \sigma^2_{\epsilon_m})$, for $m = 1, 2$. The standard deviations of the

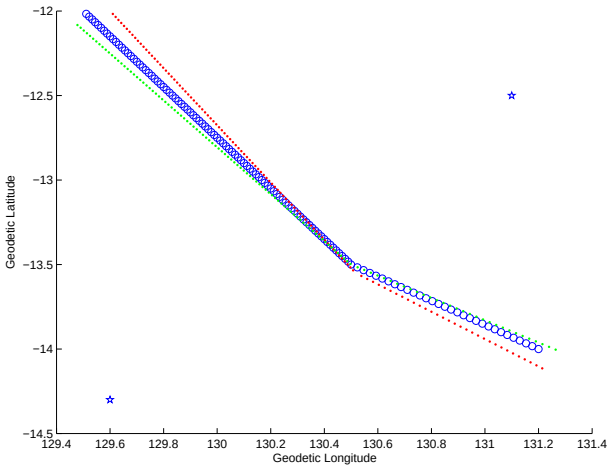


Fig. 10. True (circle) and biased (dot) target trajectories estimated by two sensors (star)

measurement noise used in the simulations are [31]: $\sigma_{\rho_1} = \sigma_{\rho_2} = 100$ m; $\sigma_{\gamma_1} = \sigma_{\gamma_2} = 0.2^o$ and $\sigma_{\epsilon_1} = \sigma_{\epsilon_2} = 0.25^o$.

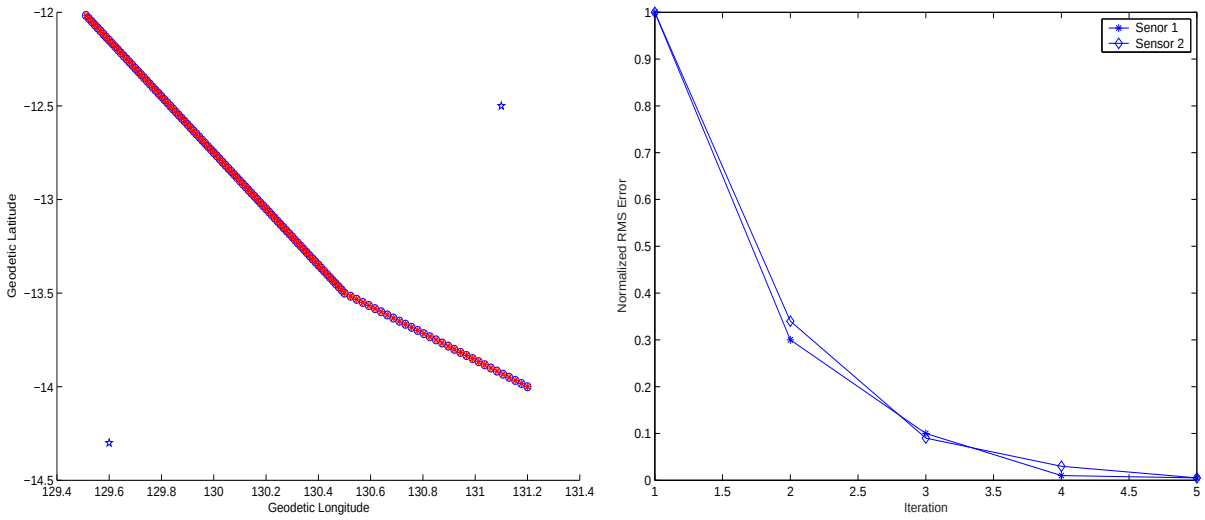
Figure 10 shows the true trajectory of the target as well as the initial estimates of the target position by the two sensors in geodetic coordinates. The differences between these trajectories are the result of unknown bias values for the sensors. Figure 11a shows these trajectories after the application of the EM-PF algorithm for sensor registration. It can be seen from this figure that the EM-PF algorithm is capable of compensating for the effect of the bias errors in track estimation after four iterations.

Figure 11b shows the RMS error of the position estimation for 100 Monte Carlo runs of the EM-PF algorithm. The algorithm is successful in compensating the effect of the unknown bias terms existing in the two sensors. It can be seen from the figure that the estimation error converges to a small value after four iterations.

The performance of the EM–PF algorithm is virually identical to that of the MLR method, shown in [31] for the same simulation scenario. However, unlike the MLR method, the EM–PF algorithm can be applied to non–Gaussian noise. Further, the EM–PF method is general technique, applicable to a wide range of problems, which include linear or nonlinear models and Gaussian or non–Gaussian noise in the presence of model uncertainty.

6. Conclusions

An EM-type algorithm for solving a joint estimation-identification problem for nonlinear non-Gaussian state-space estimation when the observation model is uncertain, is proposed. The expectation (E) step is implemented by a particular type of particle filter that is initialized by a Monte-Carlo Markov chain



(a) (b)

Fig. 11. (a)True and registered target trajectories after application of the EM-PF algorithm, and (b) RMS position error for two sensors vs. iteration number

algorithm. Within this step, the posterior distribution of states given the measurements as well as the state vectors are estimated. Consequently, in the maximization (M) step, the nonlinear measurement process parameters are approximated using a nonlinear regression method for adjusting the parameters of a mixture of Gaussians (MofG) model. The model parameters are determined by solving a linear system of equations. The proposed method, which we refer to as the EM-PF algorithm, is used to solve a highly nonlinear bearing-only tracking problem with uncertain (biased) measurements. It is shown that the algorithm is capable of accurately tracking the state vector while identifying the unknown measurement dynamics. Also, the EM-PF algorithm is applied to solving a sensor registration problem in a multisensor fusion scenario. It is shown that the algorithm is successful in compensating the effect of unknown bias terms existing in the sensors in the target tracking scenario.

By using a nonlinear regression method based on fitting a mixture of Gaussians to the observations, the algorithm is capable of approximating a wide range of nonlinearities in the measurement and state transition processes. Also, implementing the E-step with a particle filter provides the possibility of employing the algorithm in the presence of non-Gaussian noise, e.g., with impulsive or multi-modal distributions.

APPENDIX

Here we wish to evaluate

$$\min_{\boldsymbol{\theta},\boldsymbol{Q}} \sum_{t=1}^{L} \int_{\chi} p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]d\boldsymbol{x} + \ln|\boldsymbol{Q}| \tag{60}$$

A. Solution for $\boldsymbol{\theta}_k$

The problem at hand is equivalent to solving

$$\frac{\partial}{\partial \boldsymbol{\theta}} \sum_{t=1}^{L} \int_{\chi} p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\boldsymbol{\theta}\boldsymbol{\Phi}_k(t)]d\boldsymbol{x} + \ln|\boldsymbol{Q}| = \boldsymbol{0}. \tag{61}$$

In taking the expectations, we assume $\boldsymbol{\theta}_k$ in $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)$ is held fixed at the value obtained in the previous iteration [18]. Also, since $\boldsymbol{\theta}$ is independent of $\boldsymbol{x}$, we can move the derivative operator inside the expectation. Using the relation [32]

$$\frac{\partial}{\partial \boldsymbol{A}}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})^H \boldsymbol{W}(\boldsymbol{z}-\boldsymbol{A}\boldsymbol{s}) = -2\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{A}\boldsymbol{s})\boldsymbol{s}^H$$

(61) becomes

$$\sum_{t=1}^{L} \Big\langle -2\boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]\boldsymbol{\Phi}_k^H(t) \Big\rangle = \boldsymbol{0}$$

where the angular brackets denote expectation with respect to the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)$ and $\hat{\boldsymbol{\theta}}$ is the desired estimate of $\boldsymbol{\theta}$. This is equivalent to

$$\sum_{t=1}^{L} \Big\langle [\boldsymbol{z}(t)-\hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]\boldsymbol{\Phi}_k^H(t) \Big\rangle = \boldsymbol{0}. \tag{62}$$

Eq. (62) leads to

$$\sum_{t=1}^{L} \Big\langle \boldsymbol{z}(t)\boldsymbol{\Phi}_k^H(t) \Big\rangle - \hat{\boldsymbol{\theta}} \sum_{t=1}^{L} \Big\langle \boldsymbol{\Phi}_k(t)\boldsymbol{\Phi}_k^H(t) \Big\rangle = \boldsymbol{0} \tag{63}$$

from which the result for $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{k+1}$ follows. □

B. Solution for $\boldsymbol{Q}$

The problem of relevance in this case is to solve

$$\frac{\partial}{\partial \boldsymbol{Q}} \sum_{t=1}^{L} \Big\langle [\boldsymbol{z}(t)-\hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)]^H \boldsymbol{Q}^{-1}[\boldsymbol{z}(t)-\hat{\boldsymbol{\theta}}\boldsymbol{\Phi}_k(t)] \Big\rangle + \ln|\boldsymbol{Q}| = \boldsymbol{0}. \tag{64}$$

In this case, the distribution $p(\boldsymbol{x}_k(t)|\boldsymbol{z}(t),\boldsymbol{\theta}_k)$ is independent of $\boldsymbol{Q}$, and $\boldsymbol{Q}$ is independent of $\boldsymbol{x}$, so the derivative operation with respect to $\boldsymbol{Q}$ can be moved directly inside the expectation.

Using the following derivative rules [32]

$$\begin{aligned}\frac{\partial \boldsymbol{a}^H \boldsymbol{W}^{-1} \boldsymbol{b}}{\partial \boldsymbol{W}} &= -\boldsymbol{W}^{-H} \boldsymbol{a}\boldsymbol{b}^H \boldsymbol{W}^{-H} \\ \frac{\partial |\boldsymbol{W}|}{\partial \boldsymbol{W}} &= |\boldsymbol{W}| (\boldsymbol{W}^{-1})^H,\end{aligned}$$

(64) becomes

$$\begin{aligned}\sum_{t=1}^{L} \left\langle -\hat{\boldsymbol{Q}}^{-H} [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t)][\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t)]^H \hat{\boldsymbol{Q}}^{-H} \right\rangle + \hat{\boldsymbol{Q}}^{-H} &= \boldsymbol{0} \\ \hat{\boldsymbol{Q}}^{-1} \sum_{t=1}^{L} \left\langle [\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t)][\boldsymbol{z}(t) - \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t)]^H \right\rangle &= \boldsymbol{I}_{J \times J}, \end{aligned} \tag{65}$$

where $\hat{\boldsymbol{Q}}$ is the desired solution. The last line follows by postmultiplication of the line above by $\hat{\boldsymbol{Q}}^H$, and recognizing that $\hat{\boldsymbol{Q}}^{-H} = \hat{\boldsymbol{Q}}^{-1}$. By substituting (62) into (65), and distributing the sum and expectation operators amongst the individual terms, we have

$$\hat{\boldsymbol{Q}} = \boldsymbol{Q}_{k+1} = \sum_{t=1}^{L} \left\langle \boldsymbol{z}(t) \boldsymbol{z}^H(t) \right\rangle - \sum_{t=1}^{L} \left\langle \hat{\boldsymbol{\theta}} \boldsymbol{\Phi}_k(t) \boldsymbol{z}^H(t) \right\rangle \tag{66}$$

which was to be shown. □

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Application to Tracking and Navigation*. New York: John Wiley and Sons, Inc., 2001.

[2] A. Aksoy, F. Zhang, and J. W. Neilsen-Gammon, "Simultaneous state and parameter estimation with an ensemble Kalman filter for thermallt driven circulations," *http://www.met.tamu.edu/temp/altug_paper2_draft_report.pdf*, Aug. 2004.

[3] P. Mookerjee and F. Reifler, "Reduced state estimator for systems with parametric inputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 446–461, April 2004.

[4] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: A survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, p. 103123, Jan. 1996.

[5] X. R. Li and Y. Zhang, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, vol. 41, pp. 478–493, April 1996.

[6] X. R. Li and Y. Zhang, "Multiple-model estimation with variable stucture, likely-model set algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, pp. 448–466, April 2000.

[7] V. Havlena, J. Stecha, and T. Pajdia, "Smoothing preserving discontinuity based on alternative models of parameter development," in *Proceedings of the Czech Pattern Recognition Workshop*, (Prague, Czechoslovak), pp. 75–83, 1993.

[8] L. Ljung and T. Soderstorm, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[9] H. Cox, "On the estimation of state variables and parameters for noisy dynamic systems," *IEEE Transactions on Automatic Control*, vol. 9, pp. 5–12, Jan. 1964.

[10] D. T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Transactions on Automatic Control*, pp. 434–439, Oct 1965.

[11] C. G. Hilborn and D. G. Lainiotis, "Optimal estimation in the presence of unknown parameters," *IEEE Transactions on Systems, Science, and Cybernetics*, vol. 5, pp. 38–43, Jan. 1969.

[12] A. Bagchi and P. ten Brummelhuis, “Simultaneous ML estimation of state and parameters for hyperbolic systems with noisy boundary conditions,” in *Proc. of the 29th. IEEE Conference on Decision and Control*, (Honolulu, Hi.), pp. 222–224, Dec. 5 - 7, 1990.

[13] I. Runsak, “Multiple objective optimization approach to simultaneuos identification and tracking of uncertain systems,” in *Proc. of 10th. IFAC Workshop on Control Applications of Optimization*, (Haifa, Israel), Dec. 19-21, 1995.

[14] Y. Ho and B. Whalen, “An approach to the identification and control of linear dynamics systems with unknown parameters,” *IEEE Transactions on Automatic Control*, vol. 8, pp. 255–256, July 1963.

[15] L. W. Nelson and E. Stear, “The simultaneus on-line estimation of parameters and states in linear systems,” *IEEE Transactions on Automatic Control*, vol. 21, pp. 94–98, Feb. 1976.

[16] V. Havlena, “Simultaneous parameter tracking and state estimation in a linear systems,” *Automatica*, vol. 29, pp. 1041–1052, July 1993.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, Series B 1967.

[18] J. Roweis and Z. Ghahramani, *Learning Nonlinear Dynamical Systems using the Expectation-Maximization Algorithm, in S. Haykin, Ed., Kalman Filtering and Neural Networks*. New York: Dover Publications, 2001.

[19] J. K. Tugnait, “Adaptive estimation and identification for discrete systems with Markov jump parameters,” *IEEE Transactions on Automatic Control*, vol. 27, p. 1054106, Oct 1982.

[20] J. K. Tugnait and A. H. Haddad, “Adaptive estimation in linear systems with unknown Markovian noise statistics,” *IEEE Transactions on Information Theory*, vol. 26, p. 6678, Jan. 1980.

[21] A. Zia, J. P. Reilly, S. Shirani, and T. Kirubarajan, “Stochastic EM algorithm for nonlinear state estimation with model uncertainties,” in *Proc. SPIE Conference of Signal and Data Processing of Small Targets, Volume 5204*, (San Diego, CA), pp. 240–250, Dec. 2003.

[22] A. Doucet, N. de Freitas, and N. G. Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[23] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, “Comparison of EKF, pseudomeasurement and particle filters for bearing-only target tracking problem,” in *Proc. SPIE Conference on Image and Signal Processing for Small Targets, Vol. 4728*, (Orlando, FL), pp. 240–250, Aug. 2002.

[24] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, pp. 401–422, March 2004.

[25] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: John Wiley and Sons Inc., 1997.

[26] N. G. S. Arulampalam, S. Maskell and T. Clapp, “A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[27] A. Doucet, S. Godsill, and C. Andrieu, “On Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, pp. 197–208, Oct. 2000.

[28] G. Kitagawa, “Monte calro filter and smoother for non-Gaussian nonlinear state space models,” *J. Computational and Graphical Statistics*, vol. 5, pp. 1–25, Jan. 1996.

[29] N. Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications, Dept. of Elec. Eng., Linkoping University, Linkoping, Sweden*. PhD thesis, May 1999.

[30] P. Tichavsky, C. Muravchik, and A. Nehorai, “Posterior cramer-rao bounds for discrete–time nonlinear filtering,” *IEEE Trans. Signal Procesing*, vol. 46, pp. 1386–1396, May 1999.

[31] B. Ristic and N. Okello, “Sensor registration in ECEF coordinates using the MLR algorithm,” in *Proceedings of the 6th International Conference on Information Fusion*, (Cairns, Australia), July 2003.

[32] K. B. Petersen and M. S. Pedersen, “The Matrix Cookbook, http://www2.imm.dtu.dk/pubdb/p.php?3274,” Feb 2007.