FUSION OF INFORMATION FROM BIASED SENSOR DATA BY PARTICLE FILTERING

Mónica F. Bugallo, Ting Lu and Petar M. Djurić

Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794 (USA)
phone: + 1 631 632 8423, fax: + 1 631 632 8494, email: {monica, tinglu, djuric}@ece.sunysb.edu

ABSTRACT

In this paper we address the problem of fusing information from biased sensor-data collected by a sensor network. Under the assumption that the biases of the sensors are nuisance parameters, we propose an algorithm that marginalizes them out from the estimation problem. The algorithm uses particle filtering to obtain the unknown states of the system and Kalman filtering for marginalization of the biases. We apply the proposed algorithm to the problem of target tracking using bearings-only measurements acquired by more than one sensor. The advantage of the considered method over standard particle filtering which does not assume the presence of biases is illustrated through computer simulations.

1. INTRODUCTION

There are many applications where the estimation of unknown states has to be carried out in presence of unknown biases in the available measurements [1]. This problem has already been addressed in the context of sensor networks (see [2] and the references therein). Most of the work on the subject tackles the problem by decoupling the estimation of the state from the estimation of the biases, since the original approaches consisting of augmenting the state with the bias vector are deemed computationally intractable [1]. This implies that, sequential estimation of unknown states is accomplished by bias compensation after bias estimation [2, 3].

In this paper we introduce a method for estimation of the state of the system after marginalizing the biases in the observations. The proposed approach combines the particle filtering (PF) [4] and Kalman filtering (KF) [5] methodologies. Namely, the underlying idea consists of using a particle filter that estimates the state of the system and a Kalman filter for marginalizing the biases of the sensors. The latter is done by using the concept of Rao-Blackwellization [6]. Note that besides the biases, in some situations it is also possible to marginalize some of the states. This will lead to a more efficient and accurate performance of the method, i.e., the particle filter will work on a smaller state space and will explore it more comprehensively [7].

Recall that the idea behind PF is to approximate probability density functions by discrete random measures, which are composed of M particles and weights associated to the particles. In the existing literature, the Rao-Blackwellized particle filter uses one Kalman filter per particle of the particle filter. Therefore, if the particle filter is represented by M particles, we need M Kalman filters to deal with the nuisance parameters. In the literature, this approach is also known as mixture Kalman filtering [8]. The novelty of the method proposed here is based on the idea that a single Kalman filter is used to track the biases. (We note, however, that the approach of using one Kalman filter for Rao-Blackwellization has already been explored in [9].) As a result, the computational complexity of the traditional Rao-Blackwellized particle filter is significantly reduced. The proposed method is tested in the context of data fusion for target tracking using biased bearing-only observations distorted by noise.

The remaining of the paper is as follows. Section 2 introduces the problem statement. In Section 3 we explain the proposed Rao-Blackwellized particle filter method for the biased data fusion problem. Simulation results in the context of bearings-only tracking are provided in Section 4, and finally, conclusions are outlined in Section 5.

This work has been supported by the National Science Foundation under Awards CCR-0220011 and CCF-0515246 and the Office of Naval Research under Award N00014-06-1-0012.

2. PROBLEM STATEMENT

In a sensor network, the sensors collect information about a state vector, $\boldsymbol{x}_t$, that evolves with time according to

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

where $\boldsymbol{f}(\cdot)$ is a known vector function, which, in general, may be nonlinear, and $\boldsymbol{u}_t$ is a Gaussian noise vector with zero mean. Consider that there are N sensors in the network and they provide measurements that are functions of the unknown state. We model the observations as

$$\mathbf{y}_{n,t} \quad = \quad \boldsymbol{g}_n(\boldsymbol{x}_t) + \boldsymbol{b}_n + \boldsymbol{v}_{n,t}, \tag{2}$$

where the subscript n denotes the $n-$th sensor, $\mathbf{y}_{n,t}$ is the measurement of the $n-$th sensor, $\boldsymbol{g}_n(\cdot)$ is a known vector function of the state, $\boldsymbol{b}_n$ represents the unknown bias of the $n-$th sensor, and $\boldsymbol{v}_{n,t}$ is a measurement Gaussian noise with zero mean. The objective is to track the posterior probability distribution of the state, $\boldsymbol{x}_t$, given the sensor measurements, $\mathbf{y}_{1:N,1:t}$, i.e., to obtain $p(\boldsymbol{x}_t|\mathbf{y}_{1:N,1:t})$ in the presence of the unknown biases $\boldsymbol{b}_n$, $n = 1, 2, \cdots, N$.

Before we proceed, we make some comments.

1. In (2), the biases could have been incorporated in the noise, $\boldsymbol{v}_{n,t}$, where we could have modified the mean, $E(\boldsymbol{v}_{n,t})$, by an unknown value equal to the bias of the $n-$th sensor. This, of course, is equivalent to the model given by (2). However, we keep (2) because it facilitates our presentation.
2. Here we address the problem when the sensor biases are constant with time. The proposed solution presented in the next section can be extended to scenarios when these

biases may vary with time. For example, the bias could evolve following a random walk model, i.e,

$$\boldsymbol{b}_{n,t} = \boldsymbol{b}_{n,t-1} + \boldsymbol{w}_{n,t}, \quad n = 1,2,\cdots,N. \tag{3}$$

We will show in the sequel, that the solution for constant $\mathbf{b}_n$ can straightforwardly be generalized to the problem of evolving bias as given by (3).

3. Formulation (2) can be further generalized by considering

$$\mathbf{y}_{n,t} \quad = \quad \boldsymbol{g}_n(\boldsymbol{x}_t) + \mathbf{A}_n(\boldsymbol{x}_t)\boldsymbol{b}_n + \boldsymbol{v}_{n,t},$$

where $\mathbf{A}_n$ is a matrix, which in general, may be a function of the state, $\boldsymbol{x}_t$. In order to simplify the problem we have suppressed this factor although it will be considered in future work.

4. Our method is about reducing the number of Kalman filters needed for Rao-Blackwellization. In general, the method needs as many Kalman filters as there are modes in the posterior. In this paper, we assume that the posterior has only one mode and therefore the method needs only one Kalman filter. The problem of multi-modal posteriors will be addressed elsewhere.

3. PROPOSED METHOD

We seek a solution to the state estimation problem by using PF. The theory of particle filters has been well established, and its fundamentals and important applications can be found, for example, in [4] and [10]. Recall, that PF is a methodology that "approximates" the posterior *continuous* distributions by *discrete random measures*. These measures are composed of samples (particles) generated by some importance function and weights associated to the particles that compensate for the fact that the particles were not generated from the posterior distribution. Mathematically, we express the random measure by

$$\chi_t = \left\{\boldsymbol{x}_t^{(m)}, w_t^{(m)}\right\}_{m=1}^{M},$$

where $\boldsymbol{x}_t^{(m)}$ are particles that represent the states, $w_t^{(m)}$ are their weights indicating the importance of the particles, and M is the total number of particles. PF has three important operations:

1. **Sampling**: It consists of the generation of a set of new particles that represents the support of the random measure;
2. **Weight computation**: It allows for calculation of the weights of the particles;
3. **Resampling**: It replicates the particles that have large weights and removes the ones with negligible weights. Resampling is an important operation because without it PF yields very poor results.

As mentioned in the previous section, the sensor biases are considered to be constant with time. In problems with unknown constant parameters addressed by PF, one may enforce artificial evolution of the parameters [11], use the kernel smoothing procedure from [12], exploit the auxiliary PF based method from [10], or approximate the filtering density with a predefined parametric density [13]. Most of these methods impose artificial evolution of the fixed parameters and entail a large computational complexity.

We consider that the unknown biases of the sensors are nuisance parameters, and therefore we want to marginalize them. Marginalization of unwanted parameters in this context is known as Rao-Blackwellization [6]. In the wide literature, this method is applied by attaching to each particle stream one Kalman filter which estimates the unknown biases [7]. The particles are only generated for the unknown states. That often leads to much better accuracy of the PF methodology and much more efficient exploration of the state space. For example, suppose that the dimension of the state space is four, and that there are 50 sensors. Then the total number of unknowns is 54 per time instant. Now, instead of generating samples in a state space of dimension 54, we only generate samples in a four-dimensional space and let the remaining unknowns be handled by Kalman filtering.

In this paper, we propose that the implementation of the Rao-Blackwellization is carried out by one Kalman filter. This may result in substantial computational savings. The simulation results show that the new method does not have degraded performance due to the use of only one Kalman filter.

3.1 Details of the implementation

Suppose that at time t, we have the random measure,

$$\chi_t \quad = \quad \left\{\boldsymbol{x}_{0:t}^{(m)}, w_t^{(m)} = \frac{1}{M}\right\}_{m=1}^{M}. \tag{4}$$

Note that the weights of the particles in (4) are all equal, which is due to the fact that at the end of every processing cycle, we perform resampling. Also, we assume that the Kalman filter used in our method has statistics of the biases, given by the Gaussian, $\mathcal{N}(\hat{\boldsymbol{b}}_t, \hat{\mathbf{C}}_{b_t})$.

Let the observations, $\mathbf{y}_{n,t}$, $n = 1,\cdots,N$, be stacked in a vector $\mathbf{y}_t^\top = [\mathbf{y}_{1,t}^\top\ \mathbf{y}_{2,t}^\top\ \cdots\ \mathbf{y}_{N,t}^\top]$, and let $\boldsymbol{b}^\top = [\boldsymbol{b}_1^\top\ \boldsymbol{b}_2^\top\ \cdots \boldsymbol{b}_N^\top]$ be a vector containing the biases of all the sensors.

The steps of the method comprise of the following:

1. **Particle generation.** The particles of the state are drawn from
$$\boldsymbol{x}_{t+1}^{(m)} \sim \mathcal{N}\big(\boldsymbol{f}(\boldsymbol{x}_t^{(m)}), \mathbf{C}_u\big)$$
where $\mathbf{C}_u$ is the covariance matrix of the noise $\boldsymbol{u}_t$.
2. **Computation of the weights**. The weights are found from
$$w_{t+1}^{(m)} \propto p(\mathbf{y}_{t+1}|\boldsymbol{x}_{0:t+1}^{(m)}, \mathbf{y}_{1:t})$$
where $p(\mathbf{y}_{t+1}|\boldsymbol{x}_{0:t+1}^{(m)}, \mathbf{y}_{1:t})$ is a Gaussian with mean
$$\boldsymbol{\mu}_{t+1} \quad = \quad \boldsymbol{g}(\boldsymbol{x}_{t+1}^{(m)}) + \hat{\mathbf{b}}_t \tag{5}$$
and a covariance matrix
$$\mathbf{C}_{t+1} \quad = \quad \hat{\mathbf{C}}_{b_t} + \mathbf{C}_v \tag{6}$$
with $\mathbf{C}_v$ being the covariance matrix of $\boldsymbol{v}_t$. Note that $\boldsymbol{g}(\cdot)$ is a vector function given by $\boldsymbol{g}^\top(\cdot) = [\boldsymbol{g}_1^\top(\cdot)\ \boldsymbol{g}_2^\top(\cdot)\cdots\boldsymbol{g}_N^\top(\cdot)]$.
3. **Estimation of $\mathbf{x}_{t+1}$.** The estimate of $\mathbf{x}_{t+1}$ is found from
$$\hat{\mathbf{x}}_{t+1} = \sum_{m=1}^{M} w_{t+1}^{(m)} \mathbf{x}_{t+1}^{(m)}. \tag{7}$$

4. **Measurement update of the biases.** The estimate of the biases, $\boldsymbol{b}_{t+1}$, is obtained by

$$\mathbf{K}_{t+1} = \hat{\mathbf{C}}_{b_t}\left(\hat{\mathbf{C}}_{b_t}+\mathbf{C}_v\right)^{-1} \quad (8)$$
$$\hat{\boldsymbol{b}}_{t+1} = \hat{\boldsymbol{b}}_t+\mathbf{K}_{t+1}\left(\mathbf{y}_{t+1}-\boldsymbol{g}(\hat{\boldsymbol{x}}_{t+1})-\hat{\boldsymbol{b}}_t\right) \quad (9)$$
$$\hat{\mathbf{C}}_{b_{t+1}} = \left(\mathbf{I}-\mathbf{K}_{t+1}\right)\hat{\mathbf{C}}_{b_t}. \quad (10)$$

5. **Resampling.** The resampling is performed using the weights $w_{t+1}^{(m)}$.

4. COMPUTER SIMULATIONS

In this section we present computer simulations that illustrate the validity of our approach. We have considered the problem of tracking the kinematics of moving targets based on bearing-only biased measurements that are distorted by noise.

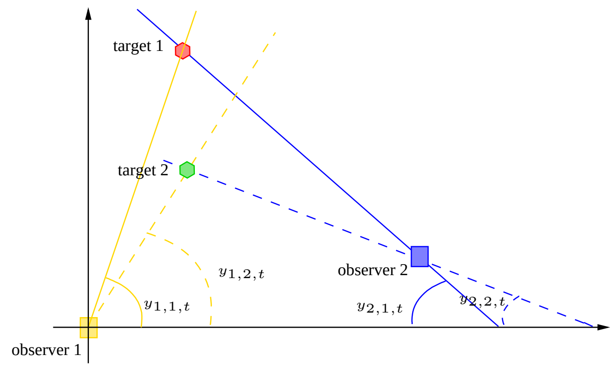

Figure 1: A system with two static sensors and their bearings-only measurements.

4.1 Bearings-only tracking problem formulation

Without loss of generality and for simplicity, we assumed that there were two targets that moved according to the following dynamic model [14][1]:

$$\boldsymbol{x}_t = \mathbf{G}_x\boldsymbol{x}_{t-1}+\mathbf{G}_u\boldsymbol{u}_t. \quad (11)$$

The system state, $\boldsymbol{x}_t^\top = [\boldsymbol{x}_{1,t}^\top \ \ \boldsymbol{x}_{2,t}^\top] \in \mathbb{R}^8$, consisted of the positions and velocities of the two targets in the field, i.e.,

$$\boldsymbol{x}_{k,t} = [x_{1,k,t} \ \ x_{2,k,t} \ \ \dot{x}_{1,k,t} \ \ \dot{x}_{2,k,t}]^\top, \quad k=1,2$$

and followed a constant velocity model governed by the state-transition matrix, $\mathbf{G}_x$ of size 8×8, and the noise-transition matrix, $\mathbf{G}_u$ of size 8×4, which were block diagonal matrices with blocks

$$\mathbf{G}'_x = \begin{pmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{G}'_u = \begin{pmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{pmatrix}$$

[1]The problem can easily be generalized to any number of targets.

where T_s was the sampling period. The state noise, $\boldsymbol{u}_t \in \mathbb{R}^4$, was due to small acceleration perturbation and was modeled as a zero mean Gaussian process with covariance matrix $\mathbf{C}_u$. Note that the model in (11) is somewhat different from that of (1) in that the state noise is multiplied by a matrix. This is reflected in the implementation of method as explained below.

The targets were tracked by two static sensors[2] positioned at $(x_{1,1}, x_{2,1})$ and $(x_{1,2}, x_{2,2})$, respectively. At time instant t, the $n-$th sensor collected an observation, $\mathbf{y}_{n,t} = [y_{1,n,t} \ y_{2,n,t}]^\top$, modeled by equation (2) where

$$\boldsymbol{g}_n(\boldsymbol{x}_t) = \begin{bmatrix} \arctan\left(\frac{x_{2,1,t}-x_{2,n}}{x_{1,1,t}-x_{1,n}}\right) \\ \arctan\left(\frac{x_{2,2,t}-x_{2,n}}{x_{1,2,t}-x_{1,n}}\right) \end{bmatrix}, \quad n=1,2,$$

$\boldsymbol{b}_n = [b_n \ b_n]^\top$ was the vector of biases for the $n-$th sensor, i.e., we assumed each sensor had only one bias[3]. The observation noise, $\boldsymbol{v}_{n,t} \in \mathbb{R}^2$ was modeled as $\mathcal{N}(\mathbf{0}, \mathbf{C}_{v_n})$.

The geometry of the problem is shown in Figure 1. Based on the made assumptions and considering that the observations are sent to a fusion center, the objective was to use the proposed algorithm to estimate the targets' locations and velocities as accurately as possible.

4.2 Implementation of the proposed algorithm

Table 1 summarizes the proposed algorithm (labeled as `PF-KF` to indicate that we used one particle filter and one Kalman filter) for the considered bearings-only tracking problem. Note that the symbol π_0 in the table denotes the prior of the target's initial state.

For comparison and benchmarking purposes, we also implemented the following algorithms:

- The standard Rao-Blackwellized particle filter that used M Kalman filters, i.e., one Kalman filter per particle, (labeled as `PF-MKF`)
- The standard particle filter that assumed complete knowledge of the biases, and therefore it did not have to estimate it (labeled as `SPF`)
- The standard particle filter that makes a wrong assumption by assuming that there were no biases (labeled as `SPFn`.)

Note that since we have more than one target, we have to deal with the problem of data association. In our approach we implemented the association by choosing the data combination that provided best fit with $\hat{\mathbf{y}}_t^{(m)\top} = [\hat{\mathbf{y}}_{1,t}^{(m)\top} \ \hat{\mathbf{y}}_{2,t}^{(m)\top}]$, $m = 1, 2, \ldots, M$, which are the estimated observations based on the particles $\boldsymbol{x}_t^{(m)}$ and $\hat{\mathbf{b}}_t$, i.e., where

$$\hat{\mathbf{y}}_{n,t}^{(m)} = g_n(\mathbf{x}_t^{(m)}) + \hat{b}_{n,t}.$$

For example, in our case of two targets, sensor $n\,(n=1,2)$ receives two observations $y_{1,n,t}$ and $y_{2,n,t}$ at time instant t, and there are two possible data combinations, $\mathbf{y}_{n,t}^1 = [y_{1,n,t} \ y_{2,n,t}]^\top$ and $\mathbf{y}_{n,t}^2 = [y_{2,n,t} \ y_{1,n,t}]^\top$. We select the

[2]Note that with two or more sensors the observability is not an issue.

[3]This assumption can be relaxed, i.e., one can assume that every measurement of the sensor has its own bias. This does not alter the algorithm except that it increases the dimensionality of the bias space.

Initialization
For $m = 1$ to M `(particle loop)`
$\mathbf{x}_0^{(m)} \sim \pi_0$
$w_0^{(m)} = \frac{1}{M}$
end
Set the values of $\hat{\boldsymbol{b}}_0$ and $\hat{\mathbf{C}}_{b_0}$

Recursive update
For $t = 1$ to T `(time loop)`
For $m = 1$ to M `(particle loop)`
Particle generation: $\boldsymbol{x}_{t+1}^{(m)} \sim \mathcal{N}\left(\mathbf{G}_x \boldsymbol{x}_t^{(m)}, \mathbf{G}_u \mathbf{C}_u \mathbf{G}_u^\top\right)$
Weight update: $\tilde{w}_{t+1}^{(m)} = \mathcal{N}\left(\boldsymbol{g}(\boldsymbol{x}_{t+1}^{(m)}) + \hat{\boldsymbol{b}}_t, \hat{\mathbf{C}}_{b_t} + \mathbf{C}_v\right)$
Weight normalization: $w_{t+1}^{(m)} = \tilde{w}_{t+1}^{(m)} / \sum_{k=1}^{M} \tilde{w}_{t+1}^{(k)}$
end
Estimation
$\hat{\boldsymbol{x}}_{t+1} = \sum_{m=1}^{M} w_{t+1}^{(m)} \boldsymbol{x}_{t+1}^{(m)}$
Bias update
$\mathbf{K}_{t+1} = \hat{\mathbf{C}}_{b_t}\left(\hat{\mathbf{C}}_{b_t} + \mathbf{C}_v\right)^{-1}$
$\hat{\mathbf{b}}_{t+1} = \hat{\mathbf{b}}_t + \mathbf{K}_{t+1}\left(\mathbf{y}_{t+1} - \boldsymbol{g}(\hat{\mathbf{x}}_{t+1}) - \hat{\mathbf{b}}_t\right)$
$\hat{\mathbf{C}}_{b_{t+1}} = \left(\mathbf{I} - \mathbf{K}_{t+1}\right)\hat{\mathbf{C}}_{b_t}$
Resampling using $w_{t+1}^{(m)}$
end

Table 1: Particle filter - Kalman filter (`PF-KF`) method for bearings-only tracking using biased measurements.

combination with the smaller error computed by

$$\epsilon^k = \frac{1}{M}\sum_{m=1}^{M} \|\hat{\mathbf{y}}_{n,t}^{(m)} - \mathbf{y}_{n,t}^k\|, \quad k = 1,2.$$

4.3 Results

We simulated evolutions of the system for $T = 300$s with a sampling period of $T_s = 1$s. The covariances of the state and observation noises were set to $\mathbf{C}_u = 0.25\mathbf{I}_4$ and $\mathbf{C}_v = 10^{-4}\mathbf{I}_4$, respectively. The coordinates of the sensors were $(x_{1,1}, x_{2,1}) = (-12000, 13000)$ m and $(x_{1,2}, x_{2,2}) = (10000, 15000)$ m, and their biases were set to $b_1 = -0.0459$ dg and $b_2 = -0.0977$ dg, respectively. In the implementation of the particle filters we used $M = 500$ particles and we set $\hat{\boldsymbol{b}}_0 = \mathbf{0}$ and $\hat{\mathbf{C}}_{b_0} = 100\mathbf{I}_2$.

Figure 2 shows the trajectories of the two targets and the obtained estimates in the two-dimensional space resulting from a single simulation of the dynamic system. It is clear that the proposed algorithm remains locked to the state trajectory.

Even though the method marginalizes the biases while estimating the nonlinear states, it can still provide estimates of the biases. Figure 3 depicts the capability of the algorithm in estimating them. The Figure also includes the evolution of the variance of the Kalman filter and, as expected, decreases with time.

Finally, we compute the average mean square error (MSE) as a performance figure of merit. It was measured in square meters and represented the difference between the true vehicle trajectories and the trajectories estimated by the particle filters,

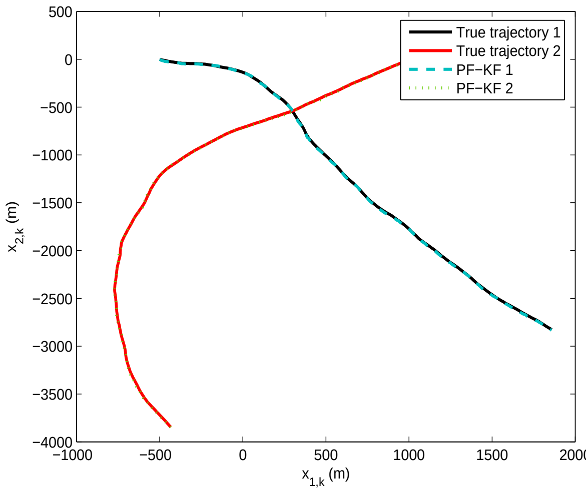

Figure 2: Trajectory of the two targets and the estimates obtained by the proposed method (`PF-KF`).

$$\text{MSE}_t = \frac{1}{4}\frac{1}{J}\sum_{k=1}^{2}\sum_{j=1}^{J}\left[(\hat{x}_{1,k,t}^j - x_{1,k,t}^j)^2 + (\hat{x}_{2,k,t}^j - x_{2,k,t}^j)^2\right]$$

where $[x_{1,k,t}^j \; x_{2,k,t}^j]^\top$ was the true position of the $k-$th target at time t in the j-th run, and $[\hat{x}_{1,k,t}^j \; \hat{x}_{2,k,t}^j]^\top$ was the corresponding estimate obtained by the filter. The MSE plots were obtained by averaging $J = 50$ independent simulations, where the trajectories in the simulations were different .

From the results shown in Figure 4, we clearly see that the worst performance was reported by the particle filter that assumed there were no biases, and that was expected. Also, it was not a surprise that the standard particle filter, which assumed complete knowledge of the bias, achieved the best performance that constituted a lower bound for the proposed method. The proposed method showed a performance close to the bound and very similar to the standard Rao-Blackwellized particle filter that used one Kalman filter per particle. Note however that the new method presented a significant computational reduction without loss of performance.

5. CONCLUSIONS

In this paper we addressed the problem of fusion of information from biased sensor measurements. We proposed a Rao-Blackwellized particle filter that uses only one Kalman filter for marginalizing the unknown biases of the sensors. The validity of the method was tested through computer simulations by applying it to a bearings-only tracking problem with two targets and two sensors. The results showed that the new method clearly outperforms the particle filter that does not assume biased sensors and is close to the performance of the standard particle filter that has complete knowledge of the biases. Furthermore, when compared to the traditional Rao-Blackwellized particle filter, it performs practically the same, while at the same time it requires much less computa-

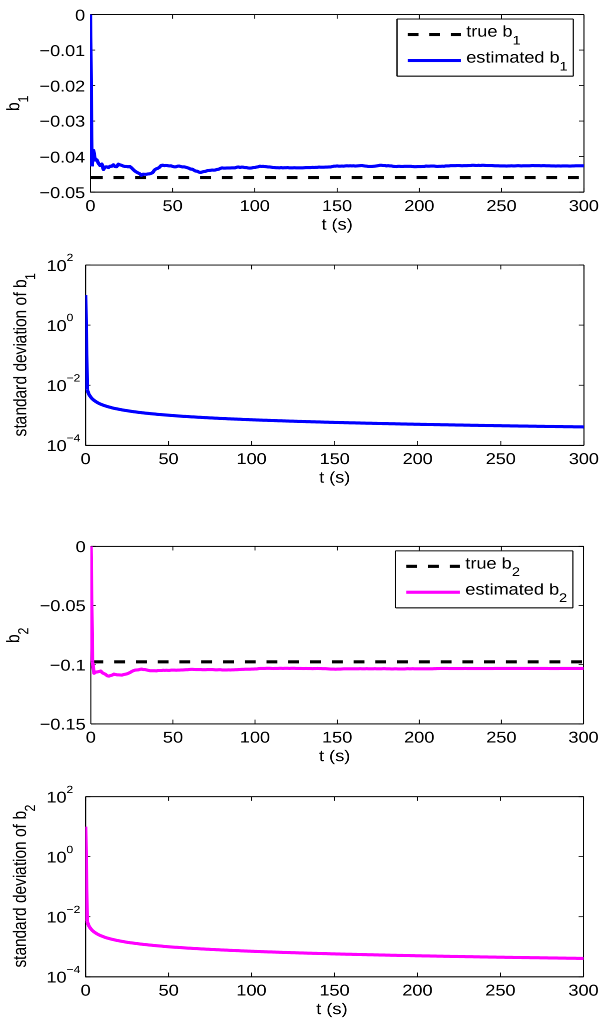

Figure 3: Means and standard deviations of the estimated biases. Top: Results corresponding to the bias of sensor 1, b_1. Bottom: Results corresponding to the bias of sensor 2, b_2.

tions. Results on the estimation of the biases suggest that the proposed method performs well too.

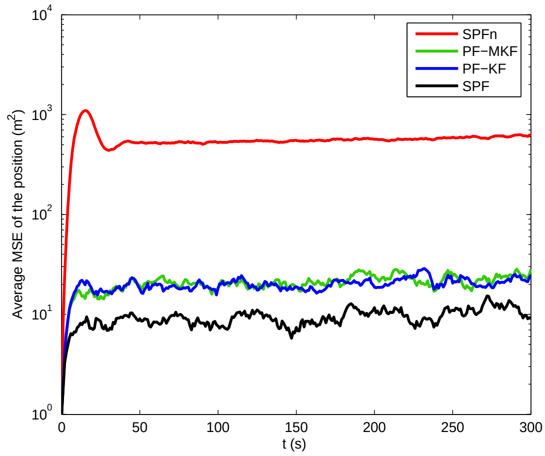

Figure 4: Average mean square errors (MSEs) in m^2 of two targets obtained by the different methods.

REFERENCES

[1] B. Friedland, “Recursive filtering in the presence of biases with irreducible uncertanty,” *IEEE Transactions on Automatic Control*, vol. 21, no. 5, pp. 789–790, 1976.

[2] X. Lin and Y. Bar-Shalom, “Multisensor target tracking performance with bias compensation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 1139–1149, 2006.

[3] X. Lin, Y. Bar-Shalom, and T. Kirubarajan, “Multisensor-multitarget bias estimation for general aynchronous sensors,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, pp. 1023–1027, 2005.

[4] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.

[5] B. Ristić, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, Boston, MA, 2004.

[6] G. Casella and C. P. Robert, “Rao-Blackwellization of sampling schemes,” *Biometrika*, vol. 83, pp. 81–94, 1996.

[7] T. Schon, F. Gustafsson, and P. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models,” *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 2279–2289, 2005.

[8] R. Chen and J. S. Liu, “Mixture Kalman filters,” *Journal of the Royal Statistical Society*, vol. 62, no. Part 3, pp. 493–508, 2000.

[9] F. Mustière, M. Bolić, and M. Bouchard, “A modified rao-blackwellised particle filter,” in *the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.

[10] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, New York, 2001.

[11] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.

[12] M. West, “Mixture models, Monte Carlo, Bayesian updating and dynamic models,” *Computer Science and Statistics*, vol. 24, pp. 325–333, 1993.

[13] P. M. Djurić, M. F. Bugallo, and J. Míguez, “Density assisted particle filters for state and parameter estimation,” in *the Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004.

[14] F. Gustaffson, F. Gunnarsson, N. Bergman, U. Forssel, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filtering for positioning, navigation, and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.