RLS-ASSISTED COST-REFERENCE PARTICLE FILTERING

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

Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794, USA
e-mail: {tinglu,monica,djuric}@ece.sunysb.edu

ABSTRACT

Cost-reference particle filtering (CRPF) allows for tracking of nonlinear dynamic states without a prior knowledge of the probability distributions of the noises in the state-space representation of the system. In this paper we consider a setup where the system unknowns consist of linear and nonlinear states. We propose an efficient scheme for estimation of the states by combining CRPF with the recursive least square (RLS) algorithm. We applied the method to the problem of target tracking using biased bearing measurements. Simulation results show a very accurate performance of the proposed approach.

***Index Terms*—** RLS, particle filtering, parameter estimation, target tracking, biased measurements

1. INTRODUCTION

Particle filtering (PF) is a sequential Monte Carlo method that approximates the posterior distributions of interest by using discrete random measures composed of particles and associated weights [1]. Based on Bayesian theory, PF proceeds in a recursive manner where each iteration consists of three major steps: (I) particle propagation (according to some importance function); (II) weight computation; and (III) resampling when needed.

Recently, a new PF scheme was proposed in [2] to deal with scenarios where the probabilistic information about the system is not available. The method, referred to as cost-reference particle filtering (CRPF) evaluates the "quality" of the particles according to some selected criterion which does not use probabilistic assumptions, unlike the weight computation in PF. The implementation of CRPF is similar to that of auxiliary PF [3].

In many systems, some states are conditionally linear, and in such cases one can exploit the property to improve the performance of the applied filter. In the context of PF, one uses the concept of Rao-Blackwellization [4], [5], which is implemented by use of Kalman filters. In this paper we use similar ideas, but for CRPF. We propose an efficient CRPF algorithm that is combined with the RLS algorithm [6]. In a way similar to the marginalized particle filters [7], only particles of the nonlinear states are generated, and the conditionally linear states are estimated by RLS. In this paper we assume that the conditionally linear states are constant with time. In the PF literature, constant states are treated in one of several ways. For example, one may use the kernel smoothing auxiliary particle filters proposed by Liu and West in [1], or apply the density assisted particle filters from [8]. We tested the proposed method in the problem of bearings-only tracking, where we assumed that the measurements were affected by biases (additive or multiplicative) and found that its performance is very good.

The paper is organized as follows. First, we formulate the problem in Section 2. In Section 3, we present the proposed method. Simulation results that demonstrate the validity of the method are provided in Section 4. Finally, Section 5 concludes the paper.

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

2. PROBLEM FORMULATION

We consider a dynamic system described by the following set of equations

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}) + \mathbf{L}(\boldsymbol{x}_{t-1})\boldsymbol{\theta} + \boldsymbol{u}_t \quad (1)$$

$$\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) + \mathbf{H}(\boldsymbol{x}_t)\boldsymbol{\theta} + \boldsymbol{v}_t \quad (2)$$

where, in (1), $\boldsymbol{x}_t$ is a nonlinear system state at time t; $\boldsymbol{f}(\cdot)$ is a known transition function, which in general may be nonlinear; $\boldsymbol{\theta}$ is a conditionally linear constant state, and $\boldsymbol{u}_t$ is a state noise vector with zero mean. The measurements, $\boldsymbol{y}_t$, are modeled by (2), where $\boldsymbol{g}(\cdot)$ represents a known measurement function, and $\boldsymbol{v}_t$ is a measurement noise with zero mean. The symbols $\mathbf{L}$ and $\mathbf{H}$ are matrices whose entries may be functions of the nonlinear states. The objective is to estimate $\boldsymbol{x}_t$ and $\boldsymbol{\theta}$, given the measurements, $\boldsymbol{y}_{1:t}$.

3. PROPOSED METHOD

In this section we first provide a brief review on CRPF and then we detail the proposed method.

3.1. The CRPF algorithm

CRPF proceeds sequentially in a similar way as auxiliary PF by updating the discrete random measure at time instant $t-1$, $\zeta_{t-1} = \left\{\boldsymbol{x}_{0:t-1}^{(m)}, c_{t-1}^{(m)}\right\}_{m=1}^{M}$, composed of particles $\boldsymbol{x}_{0:t-1}^{(m)}$ and associated costs $c_{t-1}^{(m)}$ upon the arrival of the next observation $\boldsymbol{y}_t$ [2]:

1. **Selection of streams**: Compute the risks (costs predictions) by

$$r_t^{(m)} = \lambda c_{t-1}^{(m)} + r(\boldsymbol{x}_{t-1}^{(m)}|\boldsymbol{y}_t)$$

where λ is a forgetting factor and the second term in the right-hand side of the formula measures the adequacy of the particles at time $t-1$ given the new measurement. Using the risks, resampling is carried out according to a probability mass function (pmf) defined by $\hat{\pi}_t^{(m)} \sim \mu(r_t^{(m)})$, where $\mu(\cdot)$

is a monotonically decreasing function. As a result, a new random measure is obtained

$$\tilde{\zeta}_{t-1} = \left\{ \tilde{\boldsymbol{x}}_{0:t-1}^{(m)}, \tilde{c}_{t-1}^{(m)} \right\}_{m=1}^{M}.$$

2. **Propagation of particles**:

$$\boldsymbol{x}_t^{(m)} \sim p_t(\boldsymbol{x}_t | \tilde{\boldsymbol{x}}_{t-1}^{(m)})$$

where $p_t(\cdot)$ is a probability density function (pdf) chosen by the designer.

3. **Update of the costs**:

$$c_t^{(m)} = \lambda \tilde{c}_{t-1}^{(m)} + \triangle c_t^{(m)}$$

where $\triangle c(\cdot)$ is an incremental cost, which can take different forms. For example, $\triangle c_t^{(m)} = ||\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t^{(m)}||^q$ where $\hat{\boldsymbol{y}}_t^{(m)}$ is an estimate of the observation based on the particle $\boldsymbol{x}_t^{(m)}$ and $q > 0$ [2].

4. **Estimation of the state**: One possible estimate is:

$$\hat{\boldsymbol{x}}_t = \sum_{m=1}^{M} \pi_t^{(m)} \boldsymbol{x}_t^{(m)}$$

with $\pi_t^{(m)} \propto \mu(c_t^{(m)})$. Other estimators have been discussed in [9].

3.2. The RLS-CRPF algorithm

The conventional CRPF algorithm was designed for tracking dynamic states only, and therefore the corresponding state-space model does not contain the second terms on the right-hand sides in (1)-(2). Note that since the constant state $\boldsymbol{\theta}$ is conditionally linear given $\boldsymbol{x}_t$, we are able to compute the least square (LS) estimator sequentially using the recursive LS (RLS) algorithm. Specifically, we interleave the RLS and CRPF algorithms in such a way that we run the RLS algorithm for each particle sequentially, and the obtained LS estimates are then used for resampling, propagating the particles and computing the costs. The resulting algorithm proceeds as follows (note that we denote the values of $\boldsymbol{\theta}$ in the $m-$th stream at time instant t by $\boldsymbol{\theta}_t^{(m)}$, which does not imply that $\boldsymbol{\theta}$ is a time-varying state):

1. **Selection of streams**: The risks are computed according to

$$r_t^{(m)} = \lambda c_{t-1}^{(m)} + r(\boldsymbol{x}_{t-1}^{(m)} | \boldsymbol{\theta}_{t-1}^{(m)}, \boldsymbol{y}_t)$$

and resampling is performed to obtain the new random measure

$$\tilde{\zeta}_{t-1} = \left\{ \tilde{\boldsymbol{x}}_{0:t-1}^{(m)}, \tilde{c}_{t-1}^{(m)}, \tilde{\boldsymbol{\theta}}_{t-1}^{(m)}, \tilde{\mathbf{P}}_{t-1}^{(m)} \right\}_{m=1}^{M}.$$

2. **Propagation of particles**:

$$\boldsymbol{x}_t^{(m)} \sim p_t(\boldsymbol{x}_t | \tilde{\boldsymbol{x}}_{t-1}^{(m)}, \tilde{\boldsymbol{\theta}}_{t-1}^{(m)})$$

3. **Prediction of $\boldsymbol{\theta}$ (RLS)**:

$$\begin{aligned}
\tilde{\mathbf{Q}}_{x_{t-1}} &= \mathbf{Q}_{x_{t-1}} + \mathbf{L}(\tilde{\boldsymbol{x}}_{t-1}^{(m)}) \, \tilde{\mathbf{P}}_{t-1}^{(m)} \, \mathbf{L}(\tilde{\boldsymbol{x}}_{t-1}^{(m)})^\top \\
\mathbf{G}_{t-1}^{(m)} &= \tilde{\mathbf{P}}_{t-1}^{(m)} \mathbf{L}(\tilde{\boldsymbol{x}}_{t-1}^{(m)})^\top \tilde{\mathbf{Q}}_{x_{t-1}}^{-1} \\
\boldsymbol{\theta}_{t|t-1}^{(m)} &= \tilde{\boldsymbol{\theta}}_{t-1}^{(m)} \\
&+ \mathbf{G}_{t-1}^{(m)} \Big(\boldsymbol{x}_t^{(m)} - \boldsymbol{f}(\tilde{\boldsymbol{x}}_{t-1}^{(m)}) - \mathbf{L}(\tilde{\boldsymbol{x}}_{t-1}^{(m)}) \tilde{\boldsymbol{\theta}}_{t-1}^{(m)} \Big) \\
\mathbf{P}_{t|t-1}^{(m)} &= \big(\mathbf{I} - \mathbf{G}_{t-1}^{(m)} \, \mathbf{L}(\tilde{\boldsymbol{x}}_{t-1}^{(m)}) \big) \tilde{\mathbf{P}}_{t-1}^{(m)}
\end{aligned}$$

4. **Update of the costs**:

$$c_t^{(m)} = \lambda \tilde{c}_{t-1}^{(m)} + \triangle c(\boldsymbol{x}_t^{(m)} | \boldsymbol{\theta}_{t|t-1}^{(m)}, \boldsymbol{y}_t)$$

5. **Estimation of the state**:

$$\hat{\boldsymbol{x}}_t = \sum_{m=1}^{M} \pi_t^{(m)} \boldsymbol{x}_t^{(m)}$$

6. **Update of $\boldsymbol{\theta}$ (RLS)**:

$$\begin{aligned}
\tilde{\mathbf{Q}}_{y_t} &= \mathbf{Q}_{y_t} + \mathbf{H}(\boldsymbol{x}_t^{(m)}) \, \mathbf{P}_{t|t-1}^{(m)} \, \mathbf{H}(\boldsymbol{x}_t^{(m)})^\top \\
\mathbf{K}_t^{(m)} &= \mathbf{P}_{t|t-1}^{(m)} \mathbf{H}(\boldsymbol{x}_t^{(m)})^\top \tilde{\mathbf{Q}}_{y_t}^{-1} \\
\boldsymbol{\theta}_t^{(m)} &= \boldsymbol{\theta}_{t|t-1}^{(m)} \\
&+ \mathbf{K}_t^{(m)} \Big(\boldsymbol{y}_t - \boldsymbol{g}(\boldsymbol{x}_t^{(m)}) - \mathbf{H}(\boldsymbol{x}_t^{(m)}) \boldsymbol{\theta}_{t|t-1}^{(m)} \Big) \\
\mathbf{P}_t^{(m)} &= \big(\mathbf{I} - \mathbf{K}_t^{(m)} \, \mathbf{H}(\boldsymbol{x}_t^{(m)}) \big) \mathbf{P}_{t|t-1}^{(m)}
\end{aligned}$$

7. **Estimation of $\boldsymbol{\theta}$**:

$$\hat{\boldsymbol{\theta}}_t = \sum_{m=1}^{M} \pi_t^{(m)} \boldsymbol{\theta}_t^{(m)}$$

Initialization of the algorithm includes the drawing of particles from a prior distribution, i.e., $\boldsymbol{x}_0^{(m)} \sim p_0(\boldsymbol{x}_0)$, setting the costs to zero, $c_0^{(m)} = 0$, the conditionally linear states to some initial value $\hat{\boldsymbol{\theta}}_0$, $\mathbf{P}_0^{(m)} = \alpha \, \mathbf{I}$, and the weighting matrices $\mathbf{Q}_{x_t}$, $\mathbf{Q}_{y_t}$ to matrices chosen by the designer.

4. COMPUTER SIMULATIONS

In this section we apply the proposed method to the problem of target tracking using bearings-only biased measurements.

4.1. The tracking model

The target moves according to the constant velocity motion model [10], that is,

$$\boldsymbol{x}_t = \mathbf{G}_x \boldsymbol{x}_{t-1} + \mathbf{G}_u \boldsymbol{u}_t \tag{3}$$

where $\boldsymbol{x}_t = [x_{1,t} \quad x_{2,t} \quad \dot{x}_{1,t} \quad \dot{x}_{2,t}]^\top \in \mathbb{R}^4$ is the system state representing the positions and velocities of the target; $\mathbf{G}_x$ and $\mathbf{G}_u$ are the state-transition and the noise-transition matrices, respectively, given by:

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

where T_s is the sampling period; and $\boldsymbol{u}_t \in \mathbb{R}^2$ is the state noise, which accounts for small acceleration perturbations.

Two static sensors are used to collect the bearing information of the target. Figure 1 depicts the geometry of the problem. We define the measurement function $\boldsymbol{g}(\cdot)$ as [11]

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

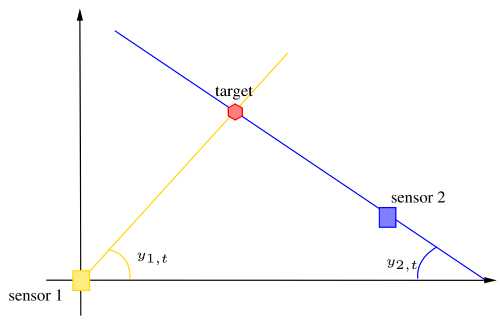

Fig. 1. A system with two static sensors and their bearings-only measurements.

where $(l_{1,1}, l_{2,1})$ and $(l_{1,2}, l_{2,2})$ denote the positions of sensors 1 and 2. We consider two different cases according to the nature of the biases: additive biases $\boldsymbol{b}_a$ and multiplicative biases $\boldsymbol{b}_m$. The observation vectors $\boldsymbol{y}_t = [y_{1,t} \;\; y_{2,t}]^\top$ for the two different cases were given by

$$\boldsymbol{y}_t = \boldsymbol{g}(\boldsymbol{x}_t) + \boldsymbol{b}_a + \boldsymbol{v}_t \qquad (4)$$

$$\boldsymbol{y}_t = \text{Diag}\Big(\boldsymbol{g}(\boldsymbol{x}_t)\Big)\boldsymbol{b}_m + \boldsymbol{v}_t \qquad (5)$$

where $\boldsymbol{v}_t \in \mathbb{R}^2$ is the observation noise and is assumed to be independent from $\boldsymbol{u}_t$. $\text{Diag}\Big(\boldsymbol{g}(\boldsymbol{x}_t)\Big)$ is the diagonal matrix with $\boldsymbol{g}(\boldsymbol{x}_t)$ as the vector of the diagonal entries. Assuming that the measurements are sent to a fusion center, the objective is to successfully track the trajectory of the target by applying the proposed method.

4.2. Results

In addition to the proposed method (labeled as `RLS-CRPF`), we also implemented, for comparison purposes, a standard CRPF with complete knowledge of the biases (labeled as `CRPF`) and a standard CRPF with the wrong assumption that there were no biases (labeled as `CRPFn`).

In the experiment, we simulated a trajectory of $T = 300$s with a sampling period of $T_s = 1$s. The target started from the origin with zero initial velocity. The two static sensors were placed at $(l_{1,1}, l_{2,1}) = (-12000, 13000)$ and $(l_{1,2}, l_{2,2}) = (10000, 15000)$, respectively (the units are meters). In the case of additive biases, we set $\boldsymbol{b}_a = [0.58 \;\; -0.96]^\top$, and for the case of multiplicative biases we considered $\boldsymbol{b}_m = [1.02 \;\; 0.99]^\top$. The state noise $\boldsymbol{u}_t$ and the observation noise $\boldsymbol{v}_t$ were modeled as mixture Gaussian distributions

$$\boldsymbol{u}_t \sim .3\mathcal{N}(0, \mathbf{I}_2) + .5\mathcal{N}(0, .25\mathbf{I}_2) + .2\mathcal{N}(0, .01\mathbf{I}_2)$$

$$\boldsymbol{v}_t \sim .5\mathcal{N}(0, 10^{-2}\mathbf{I}_2) + .4\mathcal{N}(0, 10^{-4}\mathbf{I}_2) + .1\mathcal{N}(0, 10^{-5}\mathbf{I}_2).$$

In the implementation of the filters, we used $M = 500$ particles, and we propagated the particles with a Gaussian density $\mathcal{N}(0, 4\mathbf{I}_2)$. We chose $\lambda = 0$, $q = 2$, and $\mu(x) = 1/x$. For the RLS algorithm, we set $\hat{\boldsymbol{b}}_{a0} = \mathbf{0}$, $\hat{\boldsymbol{b}}_{m0} = \mathbf{1}$, and for both cases, $\alpha = 100$, and $\mathbf{Q}_{y_t} = \mathbf{I}_2$.

Figure 2 shows the trajectories of the target and the obtained estimates. It is clear that the proposed algorithm estimated the trajectories quite accurately, while the `CRPFn` had a poor performance.

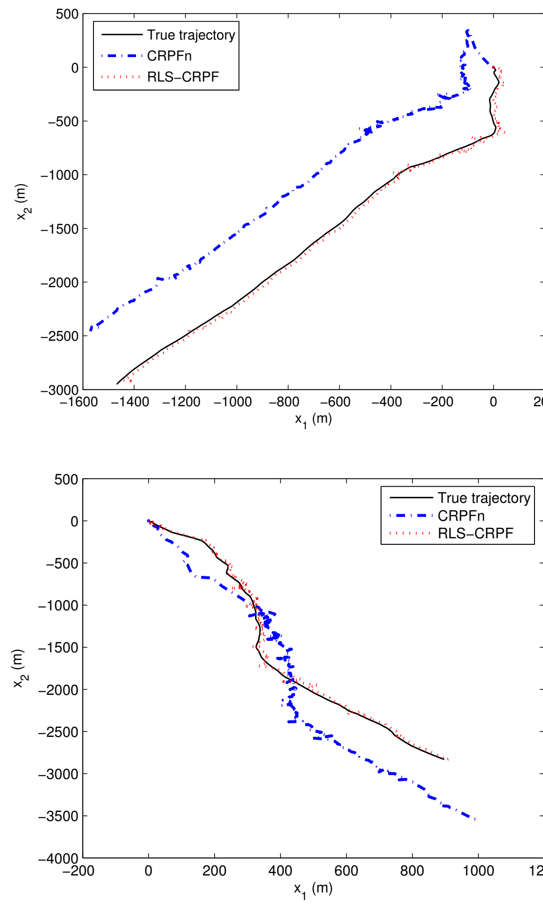

Fig. 2. Trajectory of the target and the corresponding estimates. Top: Results corresponding to additive biases. Bottom: Results corresponding to multiplicative biases.

We also provide the estimates of the biases, which are displayed in Figure 3. Again, the results show a good performance of the proposed approach.

The mean square errors (MSEs) of the position of the target are compared in Figure 4. The results there were obtained by averaging $J = 50$ independent simulations. We clearly see that the worst performance was achieved by `CRPFn`. The `CRPF`, which had complete knowledge of the biases, achieved the best performance, which constituted a lower bound for the alternative filters that assumed presence of biases but no knowledge of their values. The proposed method showed a performance much better than `CRPFn` and close to the bound.

5. CONCLUSIONS

In this paper we extend the original cost-reference particle filtering (CRPF) approach to the case of joint estimation of nonlinear dynamic states and conditionally linear constant states. The proposed method combines the recursive least square algorithm to find estimates of the linear states with the traditional CRPF algorithm which deals with the nonlinear states. This allows the CRPF algorithm to be more efficient in exploring the space of the nonlinear states. Al-

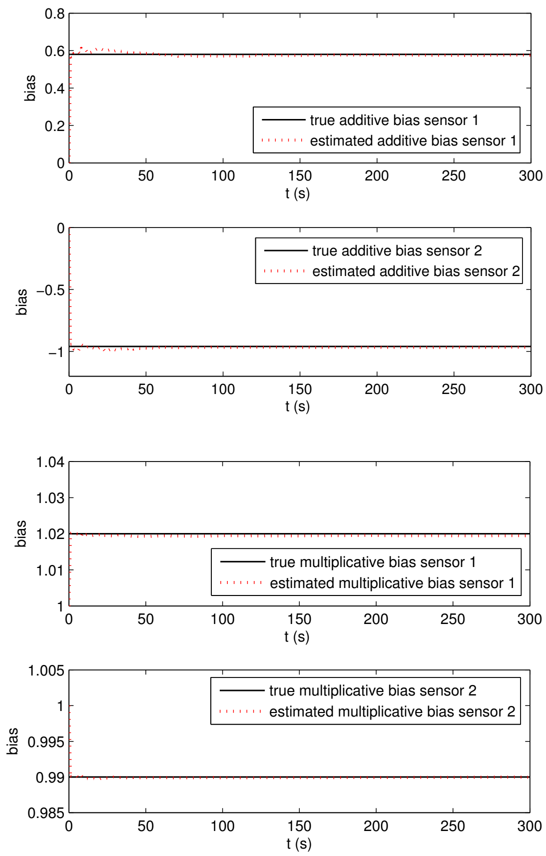

Fig. 3. Estimates of the biases. Top: Results corresponding to additive biases. Bottom: Results corresponding to multiplicative biases.

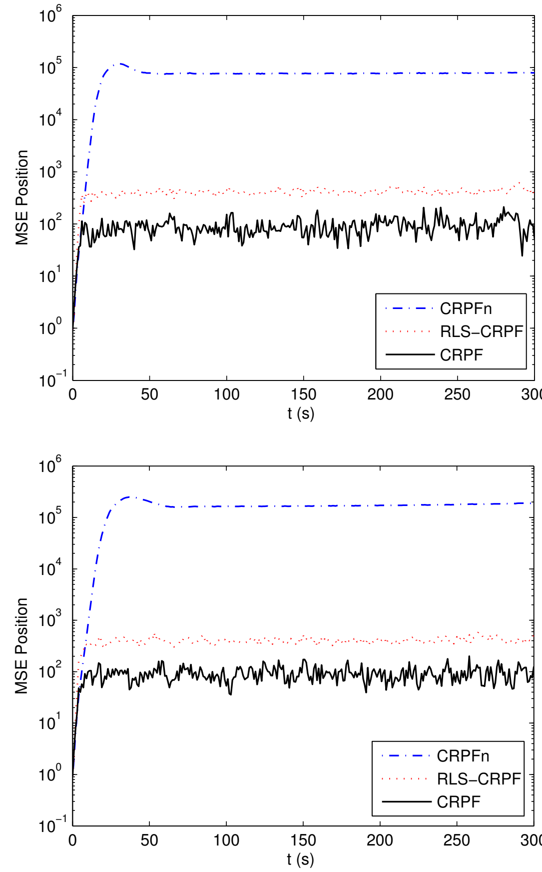

Fig. 4. MSE in m^2 for the position of the target obtained by the different methods. Top: Results corresponding to additive biases. Bottom: Results corresponding to multiplicative biases.

though the proposed method focuses on the case that conditionally linear states were constant with time, the method can be readily extended to deal with conditionally linear states that are time-varying. Simulation results on the problem of bearings-only tracking in the presence of biased measurements show very good performance of the proposed method.

6. REFERENCES

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

[2] J. Míguez, M. F. Bugallo, and P. M. Djurić, "A new class of particle filters for random dynamical systems with unknown statistics," *EURASIP Journal on Applied Signal Processing*, vol. 2004(15), pp. 2278–2294, 2004.

[3] M. Pitt and N. Shepard, "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, June 1999.

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

[5] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, New York, 1999.

[6] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, 1993.

[7] T. Schön, 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] P. M. Djurić, M. F. Bugallo, and J. Míguez, "Density assisted particle filters for stte and parameter estimation," in *the Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004.

[9] M. F. Bugallo, J. Míguez, and P. M. Djurić, "Positioning by cost reference particle filters: study of various implementations," in *Proceedings of the 2005 International Conference on "Computer as a tool", EUROCON*, Belgrade, Serbia and Montenegro, 2005.

[10] 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.

[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.