14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

ADVANCES IN COST-REFERENCE PARTICLE FILTERING

Mónica F. Bugallo 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, djuric}@ece.sunysb.edu

ABSTRACT

Recently, we have proposed a particle filtering-type methodology, which we refer to as cost-reference particle filtering (CRPF). Its main feature is that it is not based on any particular probabilistic assumptions regarding the studied dynamic model. The concepts of particles and particle streams, however, are the same in CRPF as in standard particle filtering (SPF), but the probability masses of the particles are replaced with user defined costs. In this paper we propose some modifications of the original CRPF methodology. The changes allow for development of simpler algorithms, which may also be less computationally intensive and possibly more robust. We investigate several variants of CRPF and compare them with SPF. The advantages and disadvantages of the considered algorithms are illustrated and discussed through computer simulations of tracking of multiple targets which move along a two-dimensional space.

1. INTRODUCTION

Cost Reference Particle Filtering (CRPF) is a generalization of standard particle filtering (SPF) in that it allows for recursive estimation of unobserved states of dynamic systems *without use of probability distributions of the noises in the system and prior distributions of the states* [7]. This methodology has already been successfully used in several applications including the positioning and tracking of one single target [2]. We have found that it has substantial advantages in terms of simplicity and robustness when compared to SPF algorithms [1, 2, 7].

In this paper, we continue the study of this class of PF methods and we investigate several variants of them. In particular, we propose to modify some of the steps of the original CRPF and make the resulting methods much simpler for use (by the practitioner) and much less computationally intensive. We achieve all this without degrading the performance of the CRPF methods.

We tested the methods by applying them to the problem of tracking of multiple targets by using sensor measurements obtained in a sensor network. The measurements represent a superposition of signals that carry information about the positions of the various targets. The sensors send the sensed information to a fusion center that combines the received data from all the sensors and carries out necessary computations. We chose this problem because it is challenging and highly nonlinear, and because it has already been addressed by SPF [3, 5, 6].

The remaining of the paper is organized as follows. First, in Section 2, we briefly review the basic features of CRPF and then describe the new algorithms. In Section 3, we provide the details of the multiple target tracking problem that is used for testing of the methods. In Section 4, we demonstrate their performance through computer simulations. Finally, in Section 5 we provide some concluding remarks.

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. CRPF AND ITS VARIANTS

We have a dynamic model of a system described by state and observation equations whose general forms are given by

$$\mathbf{x}_t = f_x(\mathbf{x}_{t-1}) + \mathbf{u}_t \quad (1)$$
$$\mathbf{y}_t = f_y(\mathbf{x}_t) + \mathbf{v}_t \quad (2)$$

where $t = 1, 2, \ldots$ denotes discrete time; $\mathbf{x}_t$ and $\mathbf{y}_t$ are the state and observation vectors, respectively; f_x and f_y are the state-transition and observation functions, respectively; and $\mathbf{u}_t$ and $\mathbf{v}_t$ are independent noise processes. We are interested in estimating the state vectors $\mathbf{x}_{0:t} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t]$ from the observations $\mathbf{y}_{1:t} = [\mathbf{y}_1, \ldots, \mathbf{y}_t]$. We address nonlinear problems, that is, problems where at least the observation function $f_y(\cdot)$ is nonlinear. Although, in this statement, we assumed that the noise is additive, the CRPF methods can be implemented in a similar way for other types of signal degradation.

The SPF methods are based on the use of the Bayes' theory and the knowledge of the noise distributions of the noises in (1) and (2). In many scenarios the knowledge of the distributions is questionable at best, and so our goal with CRPF was to track unknown states in time without using probabilistic assumptions of the noise. In order to estimate $\mathbf{x}_{0:t}$ from $\mathbf{y}_{1:t}$ in such situations, we considered a user-defined real *cost* function [7],

$$\mathscr{C}(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}, \lambda) = \lambda \mathscr{C}(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) + \triangle\mathscr{C}(\mathbf{x}_t|\mathbf{y}_t) \quad (3)$$

where $\triangle\mathscr{C}(\mathbf{x}_t|\mathbf{y}_t)$ is an *incremental cost* due to the presence of a new observation and $0 < \lambda < 1$ is a forgetting factor that weights the history of previous data. The cost measures the quality of the state signal estimates according to the available observations and its recursive structure allows for sequential updating using solely the state and observation vectors at the current time instant. We also considered a one-step *risk* function,

$$\mathscr{R}(\mathbf{x}_{t-1}|\mathbf{y}_t) = \triangle\mathscr{C}\left(f_x(\mathbf{x}_{t-1})|\mathbf{y}_t\right),$$

which acts as a prediction of the incremental cost obtained from the previous state. With these two basic concepts, the CRPF techniques are sequentially built in a way similar to that of SPF algorithms [7]. It is apparent that many implementations of the CRPF methodology can be proposed for a single problem.

In [1], the relationship between CRPF and SPF was studied (the values $\lambda = 1$ and $\lambda = 0$ provided the conditions) and several modifications of the CRPF methodology were proposed. The new algorithms avoid resampling (which is required in SPF and the original CRPF and represents a bottleneck in hardware implementations), are simpler to use, and are less computationally intensive. Here we outline the most simplified algorithm presented in [1] ($\lambda = 0$, no resampling, and estimation of the state using the particle with the minimum cost) and we discuss several directions that can be taken for further simplifications.

2.1 Simplified CRPF

Particles are initialized uniformly on a set, $X_0 \subset \mathbb{R}^{L_x}$ with zero assigned costs. Given a set of M state trajectories and associated costs up to time t, $\left\{\mathbf{x}_{0:t}^{(m)}, \mathscr{C}_t^{(m)}\right\}_{m=1}^{M}$, the grid of state trajectories is randomly propagated when $\mathbf{y}_t$ is observed by implementing the following steps:

1. **Selection of the most promising trajectories.**
   (a) For $m = 1, 2, \ldots, M$, let
   $$\mathscr{R}_{t+1}^{(m)} = \mathscr{R}(\mathbf{x}_t^{(m)} | \mathbf{y}_{t+1})$$
   where $\mathscr{R}_{t+1}^{(m)}$ is a risk function.
   (b) Sort in increasing order the $\mathscr{R}_{t+1}^{(m)}$.
   (c) Replicate the $\frac{M}{N}$ trajectories.

   A new particle filter denoted by $\left\{\hat{\mathbf{x}}_{0:t}^{(m)}, \hat{\mathscr{C}}_t^{(m)}\right\}_{m=1}^{M}$ is obtained as the result of this step [1].
2. **Propagation of new trajectories.**
   For $m = 1, \ldots, M$, let
   $$\begin{aligned} \mathbf{x}_{t+1}^{(m)} &\sim p_{t+1}(\mathbf{x} | \hat{\mathbf{x}}_t^{(m)}) \\ \mathscr{C}_{t+1}^{(m)} &= \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m)} | \mathbf{y}_t) \end{aligned}$$
   where p_{t+1} is a probability density function chosen by the designer. As mentioned before, this is a simplified version of the CRPF (note that compared to equation (3) the cost here is only updated using the second term).
3. **Estimation of the state.** Let $\mathbf{x}_{t+1}^{min} = \arg\min\{\mathscr{C}_{t+1}^{(m)}\}$, and therefore we choose as estimate of the state the particle with minimum cost.

2.2 New variants of the CRPF

Here we introduce some further simplifications and improvements of the above described algorithm (labeled as *crpf*).

1. **No risk calculation**
   A possible reduction in computational burden can be obtained by eliminating the risk step and moving the selection procedure at the end of step two, once costs are calculated. We will denote this method by $crpf_{nr}$.
2. **Other estimation procedures**
   The purpose of considering the minimum cost estimate was to avoid the calculation of a probability mass function to normalize the obtain costs and perform estimation. Other types of estimates like the weighted mean value of the samples can also be considered. The resulting algorithm (which does not calculate risks either) is symbolized by $crpf_{nr}^{mean}$. For ways of constructing the probability mass function $\pi(\cdot)$, please see [7].
3. **Exploring the state space**
   In order to explore the sample space more efficiently, we propose to sample P times using the same particle (each particle would have P children and therefore we would get MP particles in the generation step). The number of particles is brought back to M once the costs of all the particles are computed, the particles are ranked according to the costs, and the particles with smallest costs selected. This method is referred to as $crpf_{nr-P}$.

A summary of all the proposed algorithms is shown in the Table.

[1] Note that this CRPF approach avoids resampling by replacing it with simple ordering of the obtained risks in step 1 of the algorithm and replicating the corresponding $\frac{M}{N}$ (where N can be $2, 3, \cdots$) particles with particles with the lowest risks.

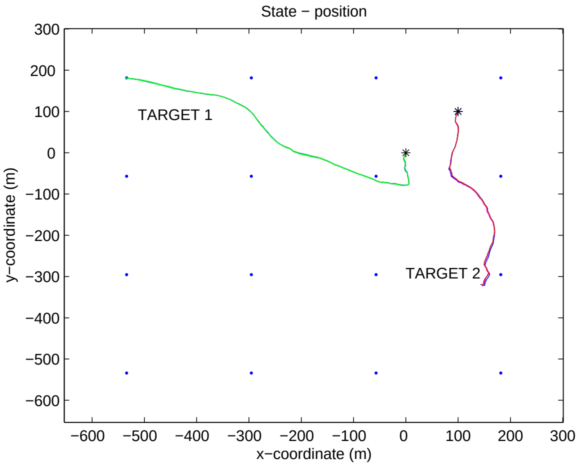

Figure 1: Target trajectories and their estimates. The estimates are very close to the true trajectories and cannot be discerned.

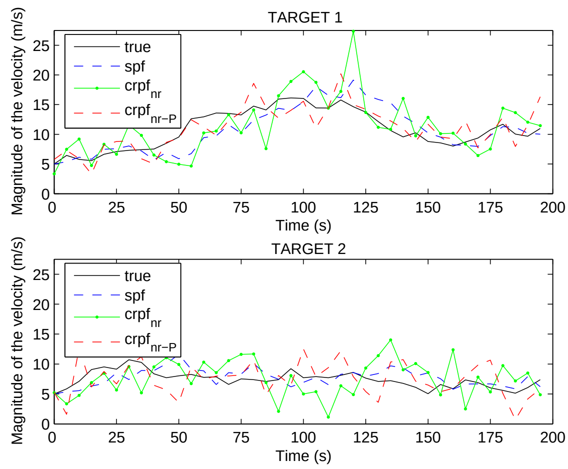

Figure 2: Magnitude of the velocity.

3. STATEMENT OF THE TRACKING PROBLEM

Consider a network of N acoustic sensors deployed in a two-dimensional field where K targets move according to a stan-

crpf	$crpf_{nr}$	$crpf_{nr}^{mean}$	$crpf_{nr-P}$
1. Select: For $m=1,\cdots,M$ $\mathscr{R}_{t+1}^{(m)} = \mathscr{R}(\mathbf{x}_t^{(m)} \mid \mathbf{y}_{t+1})$ Sort in increasing order the $\mathscr{R}_{t+1}^{(m)}$ Replicate the $\frac{M}{N}$ trajectories **2. Propagate:** For $m=1,\cdots,M$ $\mathbf{x}_{t+1}^{(m)} \sim p_{t+1}(\mathbf{x} \mid \hat{\mathbf{x}}_t^{(m)})$ $\mathscr{C}_{t+1}^{(m)} = \lambda\hat{\mathscr{C}}_t^{(m)} + \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m)} \mid \mathbf{y}_t)$ **3. Estimate:** $\mathbf{x}_{t+1}^{min} = \arg\min\{\mathscr{C}_{t+1}^{(m)}\}$	**1. Propagate:** For $m=1,\cdots,M$ $\mathbf{x}_{t+1}^{(m)} \sim p_{t+1}(\mathbf{x} \mid \hat{\mathbf{x}}_t^{(m)})$ $\mathscr{C}_{t+1}^{(m)} = \lambda\hat{\mathscr{C}}_t^{(m)} + \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m)} \mid \mathbf{y}_t)$ **2. Select:** Sort in increasing order the $\mathscr{C}_{t+1}^{(m)}$ Replicate the $\frac{M}{N}$ trajectories **3. Estimate:** $\mathbf{x}_{t+1}^{min} = \arg\min\{\mathscr{C}_{t+1}^{(m)}\}$	**1. Propagate:** For $m=1,\cdots,M$ $\mathbf{x}_{t+1}^{(m)} \sim p_{t+1}(\mathbf{x} \mid \hat{\mathbf{x}}_t^{(m)})$ $\mathscr{C}_{t+1}^{(m)} = \lambda\hat{\mathscr{C}}_t^{(m)} + \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m)} \mid \mathbf{y}_t)$ **2. Select:** Sort in increasing order the $\mathscr{C}_{t+1}^{(m)}$ Replicate the $\frac{M}{N}$ trajectories **3. Estimate:** $\mathbf{x}_{t+1}^{mean} = \Sigma_{m=1}^{M} \mathbf{x}_{t+1}^{(m)} \pi_{t+1}^{(m)}$	**1. Propagate:** For $m=1,\cdots,M$ $\mathbf{x}_{t+1}^{(m,i)} \sim p_{t+1}(\mathbf{x} \mid \hat{\mathbf{x}}_t^{(m,i)}) \quad i=1,\ldots,P$ $\mathscr{C}_{t+1}^{(m,i)} = \lambda\hat{\mathscr{C}}_t^{(m,i)} + \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m,i)} \mid \mathbf{y}_t) \quad i=1,\ldots,P$ **2. Select:** Sort in increasing order the $\mathscr{C}_{t+1}^{(m,i)}$ Replicate the $\frac{M}{N}$ trajectories **3. Estimate:** $\mathbf{x}_{t+1}^{min} = \arg\min\{\mathscr{C}_{t+1}^{(m)}\}$

dard model formulated as [5]

$$\mathbf{x}_t = \mathbf{G}_x \mathbf{x}_{t-1} + \mathbf{G}_u \mathbf{u}_t \tag{4}$$

where $\mathbf{x}_t^\top = [\mathbf{x}_{1,t}^\top, \ldots, \mathbf{x}_{K,t}^\top]^\top \in \mathbb{R}^{4K}$ indicates the position and the velocity of the targets in the field, i.e., $\mathbf{x}_{k,t} = [x_{1,k,t} \;\; x_{2,k,t} \;\; \dot{x}_{1,k,t} \;\; \dot{x}_{2,k,t}]^\top$, $k = 1, \ldots, K$.

The transition matrices, $\mathbf{G}_x$ of size $4K \times 4K$, and $\mathbf{G}_u$ of size $4K \times 2K$, are block diagonal matrices with blocks

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

where T_s is the sampling period. The noise in the state equation, $\mathbf{u}_t \in \mathbb{R}^{2K}$, accounts for small acceleration turbulences and is modeled as a Gaussian process with zero mean and covariance matrix $\mathbf{C}_u = \mathrm{diag}\left(\sigma_{u_{1,1}}^2, \sigma_{u_{1,2}}^2, \ldots, \sigma_{u_{K,1}}^2, \sigma_{u_{K,2}}^2\right)$.

The n-th sensor at a known position, $\mathbf{r}_n \in \mathbb{R}^2$, $n = 1, \ldots, N$, receives the signal power transmitted from the targets that are present in the field according to [8],

$$\begin{aligned} y_{n,t} &= g_n(\mathbf{x}_t) + v_{n,t} \qquad n = 1, \ldots, N \\ &= \sum_{k=1}^{K} \frac{\Psi_k d_0^\alpha}{\left|\mathbf{r}_n - \mathbf{l}_{k,t}\right|^\alpha} + v_{n,t}, \end{aligned} \tag{5}$$

where $g_n(\cdot)$ is a function that models the received signal power by the $n-$th sensor, Ψ_k is the emitted (known) power of the $k-$th target measured at a reference distance d_0, $\mathbf{l}_{k,t} = [x_{1,k,t}, \; x_{2,k,t}]^\top$ is the location of the $k-$the target at time t, α is an attenuation parameter that depends on the transmission medium and is considered known and the same for all sensors, and

$$\left|\mathbf{r}_n - \mathbf{l}_{k,t}\right| = \sqrt{(r_{1,n} - x_{1,k,t})^2 + (r_{2,n} - x_{2,k,t})^2}.$$

The observation noise, $v_{n,t}$, assumed to be independent from $\mathbf{u}_t$, is modeled according to $\mathscr{N}(\mu_v, \sigma_v^2)$, where $\mu_v = \sigma^2$ and $\sigma_v^2 = 2\sigma^4/L$, with σ^2 and L being the power of the background noise of one sample and the number of samples used to obtain the measured power, respectively.

The objective is to track the targets in the field, i.e., estimate $\mathbf{x}_{0:t}$, using the measurements of the N sensors, $y_{n,1:t}$, $n = 1, 2, \cdots N$.

4. SIMULATION RESULTS

We now present simulation results that show the performance of the CRPF algorithms discussed in the previous section. We ran an experiment where we generated data according to model (4)-(5), which corresponded to the evolution of a system during $T = 200$s with sampling period $T_s = 0.5$s. We considered that the distribution of the state noise was $\mathbf{u}_t \sim \mathscr{N}(\mathbf{0}, .5\mathbf{I}_{2K})$; the parameters of the observation model were, $\Psi_k = \Psi = 10^7$, $d_0 = 1$m, and $\alpha = 2$; and the observation noise distribution was generated with $\sigma^2 = 0.02$ and $L = 100$. The sensor network was composed of $N = 16$ sensors placed on a deterministic grid within the field (see Figure 1 for system configuration were the sensors are marked with dots.)

We applied the proposed CRPF methodology for solving the tracking problem and, for comparison and benchmarking purposes, we also implemented the SPF algorithm [4]. For clarity, we only included in the Figures the results corresponding to the SPF algorithm (labeled *spf*), the $crpf_{nr}$ (CRPF that does not calculate the risks - see Table), and the $crpf_{nr-P}$ (CRPF that does not calculate the risk and propagates per each particle $P = 10$ children - see Table). The *crpf* (see [1]) and the $crpf_{nr}^{mean}$ showed similar results.

We used the following cost function for the CRPF algorithms:

$$\begin{aligned} \mathscr{C}(\mathbf{x}_0) &= 0 \\ \triangle\mathscr{C}(\mathbf{x}_t \mid \mathbf{y}_t) &= \parallel \mathbf{y}_t - g(\mathbf{x}_t) \parallel \end{aligned}$$

where $\parallel \cdot \parallel$ denotes Euclidean distance. The propagation mechanism for the CRPF methods consisted of the sequence of Gaussian densities given by

$$\mathbf{x}_{t+1}^{(m)} \sim \mathscr{N}\left(\mathbf{G}_x \hat{\mathbf{x}}_{t-1}^{(m)}, \sigma_t^2 \mathbf{I}_2\right),$$

where the variance, σ_t^2, was *a priori* fixed.

In Figure 1, we compare the SPF method with perfect knowledge of the noise distributions, and we present vehicle trajectories in the two-dimensional space resulting from a single simulation of the dynamic system. Both the SPF and the CRPF-type of algorithms were run with $M = 1000$ particles. It is apparent that all the algorithms remained locked to the vehicle position during the whole simulation interval.

Figure 2 shows the magnitude of the velocity of each target in one realization of the system. Even though the curves are results of one run only, as expected, we already see that the algorithm that propagates 10 children per particle has a

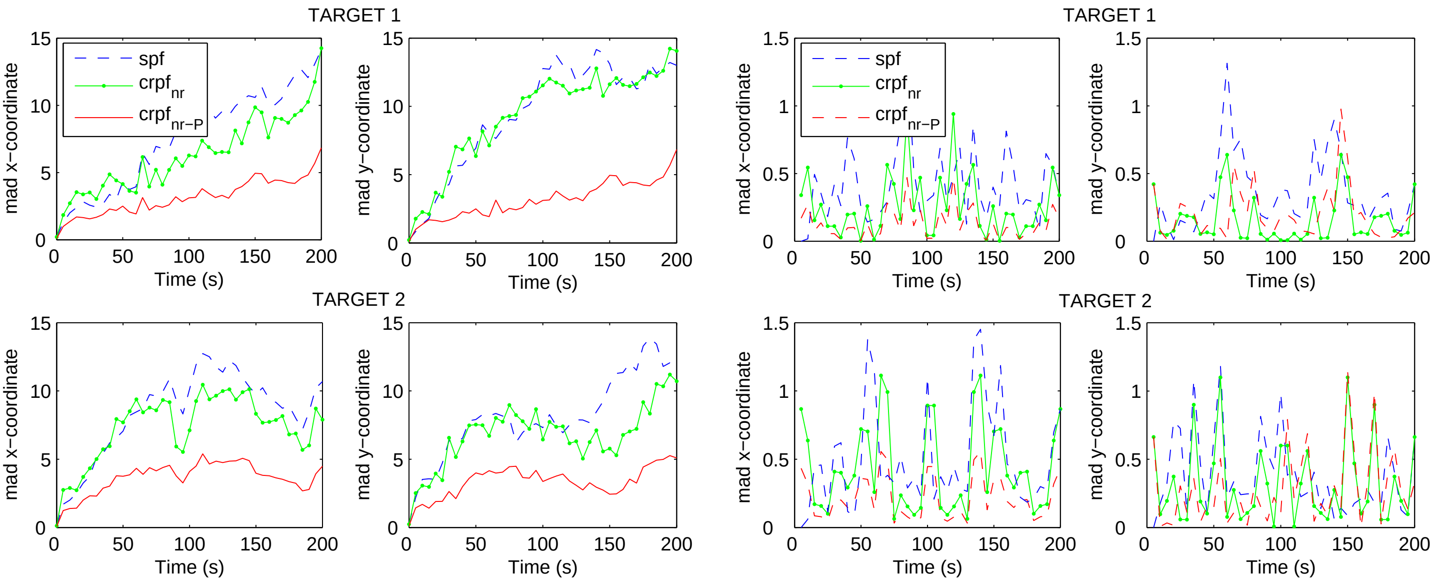

Figure 3: Left: Mean absolute deviation of the position. Right: Mean absolute deviation of the velocity.

smoother performance than the algorithm propagating only one child.

In Figure 3 (left) we show the mean absolute deviations (mad) of the estimated positions of the targets. The errors were computed over 10 runs using the same target trajectories. It can be seen that the proposed variations of the CRPF algorithm provide good tracking results. Again, note that the $crpf_{nr-P}$ that explores the sample space more intensively, gets a remarkable reductions in the error. Similar results can be seen in Figure 3 (right) where the mean absolute deviations of the estimated velocities of the targets was plotted.

5. CONCLUSIONS

We proposed three variants for simplified CRPF. Two of the modifications result in less computationally demanding algorithms, while the third approach is more computationally intensive but explores the states space better and as a result has improved performance. We studied the feasibility of the methods on the problem of tracking multiple objects which move along a certain two-dimensional area. The computer simulations showed that the new algorithms had performance that is similar to that of the SPF methods, or that it is even better.

REFERENCES

[1] M. F. Bugallo, J. Míguez, and P. M. Djurić, “Positioning by cost reference particle filters: Study of various implementations,” in *the Proceedings of the International Conference on “Computer as a tool” (EUROCON)*, Belgrade (Serbia), 2005.

[2] M. F. Bugallo, S. Xu, J. Míguez, and P. M. Djurić, “Maneuvering target tracking using cost-reference particle filtering,” in *the Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Montreal (Canada), 2004.

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

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

[5] 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.

[6] C. Hue, J.-P. Le Cadre, and P. Perez, “Sequential Monte Carlo methods for multiple target tracking and data fusion,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 309–325, 2002.

[7] 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, no. 15, pp. 2278–2294, 2004.

[8] X. Sheng and Y-H. Hu, “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 53, pp. 44–53, 2005.