Target Tracking by Particle Filtering in Binary Sensor Networks

Petar M. Djurić, *Fellow, IEEE*, Mahesh Vemula, *Student Member, IEEE*, and Mónica F. Bugallo, *Member, IEEE*

***Abstract*—We present particle filtering algorithms for tracking a single target using data from binary sensors. The sensors transmit signals that identify them to a central unit if the target is in their neighborhood; otherwise they do not transmit anything. The central unit uses a model for the target movement in the sensor field and estimates the target's trajectory, velocity, and power using the received data. We propose and implement the tracking by employing auxiliary particle filtering and cost-reference particle filtering. Unlike auxiliary particle filtering, cost-reference particle filtering does not rely on any probabilistic assumptions about the dynamic system. In the paper, we also extend the method to include estimation of constant parameters, and we derive the posterior Cramér-Rao bounds (PCRBs) for the states. We show the performances of the proposed methods by extensive computer simulations and compare them to the derived bounds.**

***Index Terms*—Binary sensor networks, particle filtering, posterior Cramér-Rao bounds (PCRBs).**

I. Introduction

RECENT advances in low-power micro sensors, actuators, embedded sensors, radio, and in general, digital wireless communication technologies have allowed development of wireless sensor networks with unparalleled capabilities [5]. Their use may span a vast range of fields, and their effectiveness is already being felt both in commercial and military applications as well as in the further development of science and engineering [6], [14], [24], [30].

In this paper, we consider wireless sensor networks with two basic components, sensors and a fusion center. The sensors sense and measure signals that provide information about an event or events of interest and send binary information to the fusion center. The fusion center combines the received information to obtain estimates about the observed phenomenon. Here the object of interest is a single target that moves in a field of sensors that measure signal power, and the objective is the online estimation of its trajectory, velocity, and power.

Often sensor networks have to operate with sensors that have limited power as well as limited communication and computational resources. At a strategic assessment workshop organized by the U.S. Army Research Lab it was concluded that [1] "it is not practical to rely on sophisticated sensors with large power supply and communication [demands]. Simple, inexpensive individual devices deployed in large numbers are likely to be the source of the battlefield awareness in the future." One type of networks that fits this description is the class of binary sensor networks, where the sensors transmit only binary information about sensed events (the event is sensed or is not sensed). The signals that reach the fusion center of these networks are therefore binary signals embedded in noise, and they pose challenging problems for recovering the sensed information by the sensors.

The sensed information in this paper is the intensity (signal power) of the transmitted signals by one or more sources that attenuates as a function of distance from the sources.[1] For the tracking we propose to apply the auxiliary particle filter (APF) [23] and the cost-reference particle filter (CRPF) [3], [16]. The two methods are sequential statistical signal processing procedures with distinct features but with similar algorithmic outline. Convergence properties of the particle filtering methods can be found in [8] and [19] and of the CRPF in [16] and [17]. The main contribution of the paper is in the application of the APF and CRPF on binary signals and in the presence of hidden complete measurements by the sensors. We show how the signals from the binary sensors contribute to producing evolving random grids of the methods, how the associated metrics of the grid nodes are updated, and how the estimates of the unknowns are obtained. We also outline the steps for computing the posterior Cramér-Rao bounds (PCRBs) of the state estimates.

The paper is organized as follows. First, in Section II we provide a brief overview of the literature related to binary sensor networks. In Section III, we describe the binary sensor network and define the addressed problem in a mathematical form. In Section IV we present the APF and CRPF algorithms for tracking a single target. In the following section, we discuss extensions of the algorithms for the case of unknown emitted power. In Section VI we address the PCRBs of the tracked variables. Details of the derivation of the bounds are shown in the Appendix. Simulations that demonstrate the performance of the proposed methods are given in Section VII. We end the paper with Section VIII where we provide final remarks.

II. A Brief Literature Survey

Binary sensors have already been addressed in the wide literature. In [15], a cooperative tracking method based on acoustic binary sensors was described. There, the tracking is based on a cooperative algorithm where the model of the target path is a

Manuscript received October 6, 2006; revised October 31, 2007. This work was supported in part by the National Science Foundation under Awards CCF-0515246 and by the Office of Naval Research under Award N00014-06-1-0012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Venkatesh Saligrama.

The authors are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: djuric@ece.sunysb.edu; vema@ece.sunysb.edu; monica@ece.sunysb.edu).

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

Digital Object Identifier 10.1109/TSP.2007.916140

[1]We note that binary sensors of other sensing modalities can be used analogously along the lines presented in this paper.

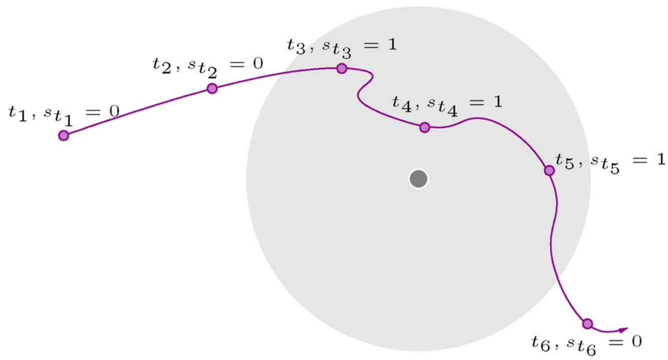


Fig. 1. A binary sensor with a target passing nearby. The signals transmitted by the sensors are denoted by s_{t_k}.

piece-wise linear curve and is different from the one that we use in our paper. In [2], as in this paper, a particle filtering-type algorithm for tracking was introduced. However, the authors focus on geometric properties of the sensors' configuration and derive their algorithm based on that geometry. Also, the method is based on sensor data that provide information about approaching or receding targets with respect to the individual sensors, whereas in our approach we only use information about the proximity of the target to the sensor. In [12], a tracking method was proposed that first estimates the positions of a target in its most recent past and then fits them with a piece-wise trajectory. In [21], another method for distributed tracking in binary sensor networks was developed. It is derived by using hidden state estimation and the Viterbi algorithm.

In other recent publications on signal processing for binary sensor networks, various problems have been addressed. For example, in [20] acoustic binary sensors were used for maximum likelihood localization (not tracking) of a source, and in [29] they were applied for system identification. Decentralized detection by binary sensors was presented in [4] where the sensors use a multiple access channel of limited capacity. Sufficient conditions were found that allow for minimal probability of error at the fusion center.

III. Network Description and Mathematical Models

A. Network Description

In a binary sensor network, the deployed sensors measure a signal of interest and if the level of the measured signal is above a predefined threshold, they report to the fusion center with a signal that identifies them; otherwise they are silent. Their function is best described by a way of example. In Fig. 1, a binary sensor is represented by the small circle and its range by the larger circle. When the target is outside the range of the sensor, the received signal is below the set threshold, and the sensor does not transmit anything (instants t_1, t_2 and t_6). During the time when the target is inside the range of the sensor, the received signal is above the threshold, and the sensor transmits a "one" to the fusion center (instants t_3, t_4, and t_5). When at a given time the fusion center does not receive a signal from a particular sensor, this implies that the sensor transmits a "zero."

The network consists of N binary sensors that may be deployed randomly, deterministically, or both. In all cases, we assume that the fusion center knows the locations of all the sensors and that the locations remain fixed for all time.

B. Mathematical Models

The model for target movement is standard and is described by [11]

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

where $\mathbf{x}_t = [x_{1,t}\ x_{2,t}\ \dot{x}_{1,t}\ \dot{x}_{2,t}]^\top \in \mathbb{R}^4$ is a state vector, which indicates the position and the velocity of the target in a two-dimensional Cartesian coordinate system, $\mathbf{G}_x$ and $\mathbf{G}_u$ are known matrices given by

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

with T_s being the sampling period and $\mathbf{u}_t$, a 2×1 zero-mean vector representing the state noise process and which accounts for the acceleration of the target. The APF method uses an *additional* assumption about $\mathbf{u}_t$; it is a *Gaussian* vector with a covariance matrix $\mathbf{C}_u = \mathrm{diag}(\sigma_{u_1}^2, \sigma_{u_2}^2)$.

The received power can be modeled as in [22] or as in [25]. We adopt the latter model, that is, the measurement of the nth sensor is given by

$$\begin{aligned} y_{n,t} &= g_n(\mathbf{x}_t) + v_{n,t} \\ &= \frac{\Psi d_0^\alpha}{\|\mathbf{r}_n - \mathbf{l}_t\|^\alpha} + v_{n,t}, \quad n = 1, 2, \cdots, N \end{aligned} \tag{2}$$

where $g_n(\cdot)$ is a function that models the received signal power by the nth sensor; $v_{n,t}$ is a noise process independent from $\mathbf{u}_t$ and independent from noise samples of other sensors; $\mathbf{r}_n \in \mathbb{R}^2$ is the position of the nth sensor; $\mathbf{l}_t = [x_{1,t}\ x_{2,t}]^\top$ is the location of the target at time t; $\|\mathbf{r}_n - \mathbf{l}_t\|$ denotes the Euclidean distance between $\mathbf{r}_n$ and $\mathbf{l}_t$; Ψ is the emitted power of the target measured at a reference distance d_0; α is an attenuation parameter that depends on the transmission medium and is considered to be known and the same for all sensors. For the application of the APF, we need to know the distribution of $v_{n,t}$. In our model, we assume that $v_{n,t} \sim \mathcal{N}(\mu_v, \sigma_v^2)$ where $\mu_v = \sigma^2$ with σ^2 being the known power of the background measurement noise of one sample and $\sigma_v^2 = 2\sigma^4/L$, with L being the number of samples used to obtain the measured power [25]. For the CRPF, we only assume that we know μ_v.

The nth sensor $(n = 1, \ldots, N)$ measures the received power $y_{n,t}$, processes it locally and transmits a single binary digit to the fusion center according to the following rule.

1) The sensor compares the actual observed power level, $y_{n,t}$, with a threshold, γ. If the sensed value is below γ, it does not transmit anything.
2) If the sensed value is greater than γ, the sensor transmits its identification code to the fusion center.

Therefore, the sensors in the network send signals to the fusion center *only* if the received power is greater than the sensor thresholds.

The received signal from the nth sensor at the fusion center is modeled as

$$z_{n,t} = \beta_n s_{n,t} + \epsilon_{n,t} \tag{3}$$

where

$$s_{n,t} = \begin{cases} 1 & \text{if } y_{n,t} > \gamma \\ 0 & \text{if } y_{n,t} \leq \gamma \end{cases} \tag{4}$$

and where $\epsilon_{n,t}$ is the observation noise, and β_n is a *known* attenuation coefficient associated with the nth sensor. Again, for the APF method we need the knowledge of the noise distribution of $\epsilon_{n,t}$, and we let $\epsilon_{n,t} \sim \mathcal{N}(0, \sigma_\epsilon^2)$, whereas for the CRPF method we only assume that the noise is zero mean.

In summary, the measurements made by the sensors are complete and are modeled by (2). The sensors, however, always transmit binary signals constructed according to (4), and the fusion center receives them as quantified by (3).

The objective is to track the evolving state $\mathbf{x}_{0:t} = (\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_t)$ using the observations $\mathbf{z}_{1:t} = (z_{1,1:t}, \ldots, z_{N,1:t})$, that is, the observations up to time instant t of the first sensor, $z_{1,1:t}$, the second sensor, $z_{2,1:t}$, as well as the remaining $N-2$ sensors, $z_{3,1:t}, \cdots, z_{N,1:t}$. The APF uses probabilistic assumptions about all the noises in the model and about the prior of the states. The CRPF only needs knowledge of the first moments of the noises.

IV. Tracking Algorithms

First we present a tracking algorithm based on APF [10], [23] (Section IV-A), and then we proceed with the presentation of a CRPF algorithm [3], [16] (Section IV-B). In this section, the parameter Ψ is assumed known, but subsequently this assumption will be dropped.

A. APF Algorithm

Recall that according to the theory of particle filtering, we track the *a posteriori* distribution of $\mathbf{x}_{0:t}$, $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$, by approximating it with a random measure χ_t composed of particles $x_t^{(m)}$ and weights $w_t^{(m)}$, where m is an index, and which we denote by $\chi_t = \{\mathbf{x}_{0:t}^{(m)}, w_t^{(m)}\}_{m=1}^M$ with M being the number of particles. At every time instant t, the particle filter carries out the following operations: (1) Selection of most promising particle streams, (2) particle propagation, (3) computation of particle weights, and (4) state estimation.

The APF attempts to draw from an importance function which is as close as possible to the optimal one. To that end, the selection of most promising particles is carried out by sampling from a multinomial distribution where the number of possible outcomes is M and the probabilities of the respective outcomes are $\tilde{w}_t^{(m)}$, $m = 1, 2, \cdots, M$, and

$$\tilde{w}_t^{(m)} \propto p\left(\mathbf{z}_t|\boldsymbol{\mu}_t^{(m)}\right) w_{t-1}^{(m)} \tag{5}$$

where $\boldsymbol{\mu}_t^{(m)}$ is some parameter that characterizes $\mathbf{x}_t^{(m)}$ given $\mathbf{x}_{t-1}^{(m)}$.

Since the noise samples $\epsilon_{n,t}$ from (3) are assumed independent, we have

$$p\left(\mathbf{z}_t|\boldsymbol{\mu}_t^{(m)}\right) = \prod_{n=1}^N p\left(z_{n,t}|\boldsymbol{\mu}_t^{(m)}\right). \tag{6}$$

For the factors $p(z_{n,t}|\boldsymbol{\mu}_t^{(m)})$, we can write

$$\begin{aligned} p\left(z_{n,t}|\boldsymbol{\mu}_t^{(m)}\right) =& p\left(z_{n,t}|s_{n,t}=0, \boldsymbol{\mu}_t^{(m)}\right) P\left(s_{n,t}=0|\boldsymbol{\mu}_t^{(m)}\right) \\ &+ p\left(z_{n,t}|s_{n,t}=1, \boldsymbol{\mu}_t^{(m)}\right) P\left(s_{n,t}=1|\boldsymbol{\mu}_t^{(m)}\right) \\ =& p(z_{n,t}|s_{n,t}=0) P\left(s_{n,t}=0|\boldsymbol{\mu}_t^{(m)}\right) \\ &+ p(z_{n,t}|s_{n,t}=1) P\left(s_{n,t}=1|\boldsymbol{\mu}_t^{(m)}\right) \end{aligned} \tag{7}$$

where

$$p(z_{n,t}|s_{n,t}) = \mathcal{N}\left(\beta_n s_{n,t}, \sigma_\epsilon^2\right) \tag{8}$$

and

$$P\left(s_{n,t}=1|\boldsymbol{\mu}_t^{(m)}\right) = Q\left(\frac{\gamma - g_n\left(\boldsymbol{\mu}_t^{(m)}\right) - \mu_v}{\sigma_v}\right) \tag{9}$$

$$P\left(s_{n,t}=0|\boldsymbol{\mu}_t^{(m)}\right) = 1 - Q\left(\frac{\gamma - g_n\left(\boldsymbol{\mu}_t^{(m)}\right) - \mu_v}{\sigma_v}\right) \tag{10}$$

where $Q(\cdot)$ denotes the complementary of the standard normal cumulative distribution function.

At the beginning, the initial set of particles $\mathbf{x}_0^{(m)}$, $m = 1, 2, \cdots, M$, are drawn from a prior distribution $\pi(\mathbf{x}_0)$, and the weights of the particles are set to $1/M$. Suppose now that at time instant $t-1$, we have the random measure $\chi_{t-1} = \{\mathbf{x}_{0:t-1}^{(m)}, w_{t-1}^{(m)}\}_{m=1}^M$. Then the steps of a particle filter recursion can be implemented as follows.

1) Selection of Most Promising Particle Streams: For selection of the most promising particles, we use the conditional mean of $\mathbf{x}_t^{(m)}$ given $\mathbf{x}_{t-1}^{(m)}$ as a characterizing parameter of every stream, i.e.

$$\boldsymbol{\mu}_t^{(m)} = E\left(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}\right). \tag{11}$$

The conditional means are computed readily from

$$\boldsymbol{\mu}_t^{(m)} = \mathbf{G}_x \mathbf{x}_{t-1}^{(m)}. \tag{12}$$

This is followed by computation of the weights according to (5) and their normalization. Finally, a set of indices $\{i_m\}$ are drawn from the probability mass function (pmf) represented by the normalized weights.

2) New Particle Generation: The first two elements of the four-dimensional state $\mathbf{x}_t$ represent the location of the target in a two-dimensional space, and the remaining elements are the components of the velocity in this space. That implies that the generation of $\mathbf{x}_t^{(m)}$ requires drawing only two-dimensional random variables. The generation can be carried out by first, propagating the velocity components one step ahead using the joint distribution $p(\dot{x}_{1,t}, \dot{x}_{2,t}|\dot{x}_{1,t-1}, \dot{x}_{2,t-1})$

or $p(\dot{x}_{1,t}, \dot{x}_{2,t} | \dot{x}_{1,t-1}, \dot{x}_{2,t-1}, \mathbf{z}_t)$ and second, computing the locations according to

$$x_{1,t}^{(m)} = x_{1,t-1}^{(i_m)} + \frac{T_s}{2}\left(\dot{x}_{1,t}^{(m)} + \dot{x}_{1,t-1}^{(i_m)}\right) \quad (13)$$

$$x_{2,t}^{(m)} = x_{2,t-1}^{(i_m)} + \frac{T_s}{2}\left(\dot{x}_{2,t}^{(m)} + \dot{x}_{2,t-1}^{(i_m)}\right). \quad (14)$$

The above equations are obtained from (1).

3) Weight Computation: The newly generated particles are assigned weights according to

$$w_t^{(m)} \propto \frac{p\left(\mathbf{z}_t | \mathbf{x}_t^{(m)}\right)}{p\left(\mathbf{z}_t | \boldsymbol{\mu}_t^{(i_m)}\right)}.$$

The likelihood terms of the numerator and denominator are calculated as in the APF using (6)–(10).

4) State Estimation: Once the weights are normalized, one can use χ_t to compute estimates of the unknown states. If we want the minimum mean square error (MMSE) estimate, we obtain it from

$$\hat{\mathbf{x}}_t = \sum_{m=1}^{M} w_t^{(m)} \mathbf{x}_t^{(m)}. \quad (15)$$

B. CRPF Algorithm

The objective of CRPF is to estimate sequentially the evolution of the unknown state $\mathbf{x}_{0:t}$ from $\mathbf{z}_{1:t}$ without assumptions about the probability distributions of the noises in the model. It is similar in structure to that of the APF because CRPF, too, uses a discrete random measure. This random measure is composed of particles and costs associated to them, where the costs are user-defined. We denote the random measure by $\zeta_t = \{\mathbf{x}_{0:t}^{(m)}, \mathcal{C}_t^{(m)}\}_{m=1}^{M}$, where $\mathbf{x}_{0:t}^{(m)}$ has the same meaning as before and $\mathcal{C}_t^{(m)}$ are the associated costs to $\mathbf{x}_{0:t}^{(m)}$. It is clear that the costs here play the role of the weights in APF. In fact, with appropriate choices of the costs, one can make the CRPF equivalent to the APF or the standard particle filter [3].

In general, the costs are updated according to [16]

$$\begin{aligned}\mathcal{C}_t^{(m)} &= \mathcal{C}\left(\mathbf{x}_{0:t}^{(m)} | \mathbf{z}_{1:t}\right) \\ &= \lambda \mathcal{C}\left(\mathbf{x}_{0:t-1}^{(m)} | \mathbf{z}_{1:t-1}\right) + \triangle\mathcal{C}\left(\mathbf{x}_t^{(m)} | \mathbf{z}_t\right)\end{aligned} \quad (16)$$

where λ is a forgetting factor $(0 \le \lambda \le 1)$, and $\Delta\mathcal{C}(\mathbf{x}_t^{(m)} | \mathbf{z}_t)$ is an incremental cost. Obviously, the value of λ controls how fast the state estimates can adapt to new values of the states. The incremental cost contributes to the total cost at time instant t and is a function of the particle value and the observation at that instant. A typical incremental cost has the form

$$\triangle\mathcal{C}\left(\mathbf{x}_t^{(m)} | \mathbf{z}_t\right) = \left\| \mathbf{z}_t - \hat{\mathbf{z}}_t^{(m)} \right\|^q \quad (17)$$

where $\hat{\mathbf{z}}_t^{(m)}$ is a function of $\mathbf{x}_t^{(m)}$, and $q > 0$. So, CRPF proceeds analogously to the APF; with the vector of observations $\mathbf{z}_t$, the discrete random measure at time instant $t-1$, $\zeta_{t-1} = \{\mathbf{x}_{0:t-1}^{(m)}, \mathcal{C}_{t-1}^{(m)}\}_{m=1}^{M}$, is updated to $\zeta_t = \{\mathbf{x}_{0:t}^{(m)}, \mathcal{C}_t^{(m)}\}_{m=1}^{M}$ to reflect accurately the possible value of the unknown state at time instant t, $\mathbf{x}_t$.

The procedure has four steps: (1) selection of most promising particle streams, (2) propagation of particles, (3) cost update, and (4) state estimation. The initialization of the method is carried out by randomly drawing initial particles from some probability density function (pdf) $\pi(\mathbf{x}_0)$ whose support includes the space of $\mathbf{x}_0$. We propose that the method is implemented as follows.

1) Selection of Most Promising Particle Streams: This step is reminiscent of the main idea of the APF, where resampling at time instant $t-1$ takes place by using measurements from time instant t. For CRPF, we define a risk function $\mathcal{R}(\mathbf{x}_{t-1}^{(m)} | \mathbf{z}_t)$, which quantifies the quality of the particle $\mathbf{x}_{t-1}^{(m)}$ given the next set of observations, $\mathbf{z}_t$. As a risk function, one can use the incremental cost, that is, in our case

$$\mathcal{R}\left(\mathbf{x}_{t-1}^{(m)} | \mathbf{z}_t\right) = \triangle\mathcal{C}\left(\hat{\mathbf{x}}_t^{(m)} | \mathbf{z}_t\right)$$

with

$$\hat{\mathbf{z}}_t^{(m)} = h\left(\hat{\mathbf{y}}_t^{(m)}\right) \quad (18)$$

$$\hat{\mathbf{y}}_t^{(m)} = g\left(\hat{\mathbf{x}}_t^{(m)}\right) + \boldsymbol{\mu}_v \quad (19)$$

$$\hat{\mathbf{x}}_t^{(m)} = G_x \mathbf{x}_{t-1}^{(m)} \quad (20)$$

where the elements of $g(\cdot)$ are defined by (2) and those of $h(\cdot)$ by (3) and (4), i.e., the elements of $h(\cdot)$ are given by

$$h\left(\hat{y}_{n,t}^{(m)}\right) = \begin{cases} \beta_n, & \hat{y}_{n,t} > \gamma \\ 0, & \hat{y}_{n,t} \le \gamma \end{cases}.$$

Once the risks are computed, they are added to their costs at $t-1$ to obtain the predicted costs, i.e.

$$\hat{\mathcal{C}}_t^{(m)} = \lambda\mathcal{C}\left(\mathbf{x}_{0:t-1}^{(m)} | \mathbf{z}_{1:t-1}\right) + \mathcal{R}\left(\mathbf{x}_{t-1}^{(m)} | \mathbf{z}_t\right). \quad (21)$$

The particles are then sorted according to their predicted costs $\hat{\mathcal{C}}_t^{(m)}$ in ascending order and the first L of the M particles are replicated $J = M/L$ times (we assume that J is an integer). In other words, each of the surviving particles will have J children at time instant t [3]. We note that in previous versions of CRPF implementations we used functions to generate pmfs, where the latter were subsequently used for resampling of the particles. With the sorting scheme, we avoid the use of such functions and classical resampling and instead, we proceed directly with removing "bad" particles. Many simulation results have shown that with this simpler and faster scheme we do not sacrifice performance.

2) Particle Propagation: For particle propagation we can use a Gaussian proposal density in a similar way as is done with APF. Note that the functional form of the Gaussian is not used for computing the costs of the particles. Also, the use of a Gaussian is not required, and we can use any other density that is centered around the particle and that produces random variables with appropriate variance, like a uniform, or a Laplace, or a Cauchy density.

If we use a Gaussian, we draw the velocities of the target with mean $\dot{\mathbf{x}}_{t-1}^{(i_m)} = [\dot{x}_{1,t-1}^{(i_m)}\ \dot{x}_{2,t-1}^{(i_m)}]^\top$ and with covariance matrix $\sigma_{t-1}^{2,(i_m)} \mathbf{I}_{2\times 2}$, where the i_ms denote indexes of sorted particles, and $\dot{\mathbf{x}}_{t-1}^{(i_m)}$'s are *surviving* particles from step 1. The variance, $\sigma_{t-1}^{2,(i_m)}$, is recursively updated by (see [16])

$$\sigma_{t-1}^{2,(i_m)} = \frac{t-2}{t-1}\sigma_{t-2}^{2,(i_m)} + \frac{\left\|\dot{\mathbf{x}}_{t-1}^{(i_m)} - \dot{\mathbf{x}}_{t-2}^{(i_m)}\right\|^2}{2(t-1)}.$$

The locations are then obtained by (13) and (14).

3) Cost Update: The cost update is performed by using (16).

4) State Estimation: The state is estimated by using the particles and the associated costs. One way of carrying out the estimation is by creating a pmf from the costs. To that end, a monotonically decreasing function $\eta(\cdot)$ that converts the set of costs into probability masses $\pi_{c_t}^{(m)}$ is defined, that is

$$\pi_{c_t}^{(m)} \propto \eta\left(\mathcal{C}_t^{(m)}\right).$$

One function that has worked well for different problems is

$$\eta\left(\mathcal{C}_t^{(m)}\right) = \frac{1}{\left(\mathcal{C}_t^{(m)} - \min(\mathcal{C}_t) + 1/M\right)^2}.$$

Once the $\pi_{c_t}^{(m)}, m = 1, 2, \cdots, M$ are computed, the estimation of $\mathbf{x}_t$ can be carried out readily.

V. Extension of the APF When Ψ is Unknown

The particle filters described in the previous section are based on the assumption that in (2) the emitted power, Ψ, of the target measured at a reference distance is *known*. When it is unknown and varies with time randomly, the presented particle filters are modified straightforwardly so that Ψ_t becomes an element of the state vector $\boldsymbol{\xi}_t$, i.e., $\boldsymbol{\xi}_t = [x_{1,t}\ x_{2,t}\ \dot{x}_{1,t}\ \dot{x}_{2,t}\ \Psi_t]^\top$, and the state-space (1) is changed to reflect the evolution of Ψ_t with time.

In our model, Ψ is constant. It is well known that a special care must be taken for the estimation of constant parameters by particle filtering. Estimation of static parameters by particle filtering has already been addressed in the literature, for example, [7], [9], and [26]. The approach that we propose here exploits the concept behind Gaussian particle filtering [13]. CRPF *does not* have the problems of the APF in estimating constant parameters because it is *not* based on the use of probability distributions.

Let the parameter Ψ be denoted by Ψ_t even though it does not change with time. Formally, we have

$$\Psi_t = \Psi_{t-1} \tag{22}$$

which implies that the prior proposal distribution of Ψ should be

$$p\left(\Psi_t|\Psi_{t-1}^{(m)}\right) = \delta\left(\Psi_t - \Psi_{t-1}^{(m-1)}\right). \tag{23}$$

The particle filter from the previous section remains the same except for the following modification: at time $t-1$, one approximates the marginal posterior of Ψ_{t-1} with a Gaussian (or some other) distribution [9]. If it is a Gaussian distribution, we compute its parameters (the mean and variance of Ψ_{t-1}) from $\Psi_{t-1}^{(m)}$ by

$$\mu_{\Psi_{t-1}} = \sum_{m=1}^{M} w_{t-1}^{(m)}\Psi_{t-1}^{(m)}$$
$$\sigma_{\Psi_{t-1}}^2 = \sum_{m=1}^{M} w_{t-1}^{(m)}\left(\Psi_{t-1}^{(m)} - \mu_{\Psi_{t-1}}\right)^2. \tag{24}$$

Then we sample from $\mathcal{N}(\mu_{\Psi_{t-1}}, \sigma_{\Psi_{t-1}}^2)$, i.e., $\Psi_t^{(m)} \sim \mathcal{N}(\mu_{\Psi_{t-1}}, \sigma_{\Psi_{t-1}}^2)$. The drawn particles $\Psi_t^{(m)}$ are particles of Ψ_t since $\Psi_t = \Psi_{t-1}$, and the remaining elements of the particle $\mathbf{x}_t^{(m)}$ are generated in the same way as in the previous section.

The CRPF method applies the same method. For finding the parameters of the Gaussian, it uses the pmf determined in step 4.

VI. PCRBs

The PCRBs provide in general the lower bound for mean square errors (MSEs) [28]. In our problem, the PCRBs represent the lower bounds of the MSEs of the estimated unknown position, velocity, and emitted power of the target. In particular, for the expected squared error of $\mathbf{x}_t$ we have the lower bound given by

$$\mathbf{C}_t \triangleq E\left\{(\hat{\mathbf{x}}_t - \mathbf{x}_t)(\hat{\mathbf{x}}_t - \mathbf{x}_t)^\top\right\} \geq \mathbf{J}_t^{-1} \tag{25}$$

where $\mathbf{J}_t$ is the filtering information matrix, whose inverse is the PCRB of $\mathbf{x}_t$.

In the context of sensor networks, the PCRBs provide insights into various issues including the following:

- the accuracy attainable in estimating the dynamics of the target;
- the effect of the deployment geometry of the sensors on the PCRB; and
- the effect of the sensing properties of the sensor on the achievable accuracy.

When the dynamics of the state evolution of the target are given by (1), the prior distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is not defined because this conditional distribution becomes singular. In [27], a recursive method for determining the PCRBs for such cases was described, and we followed that approach. We also used the same notation as in [27] to allow for the more compressed presentation that is given in the Appendix. The obtained PCRBs do not have analytical expressions in closed form but they can be computed using Monte Carlo simulation methods.

VII. Simulations

We present some computer simulations that illustrate the performances of the proposed algorithms. We considered a scenario where the examined network consisted of $N = 264$ sensors deployed in a field with dimensions $800 \times 500\ \mathrm{m}^2$. The attenuation parameter was set to $\alpha = 2.5$ and the reference power parameter to $\Psi = 5{,}000$ at $d_0 = 1$ m. The parameter Ψ was assumed unknown throughout the simulations and was also estimated. The sensing radius of the sensors, which depends on the threshold γ and the power parameter Ψ, was 25 m. The corresponding threshold was set to 2. The covariance matrix of the

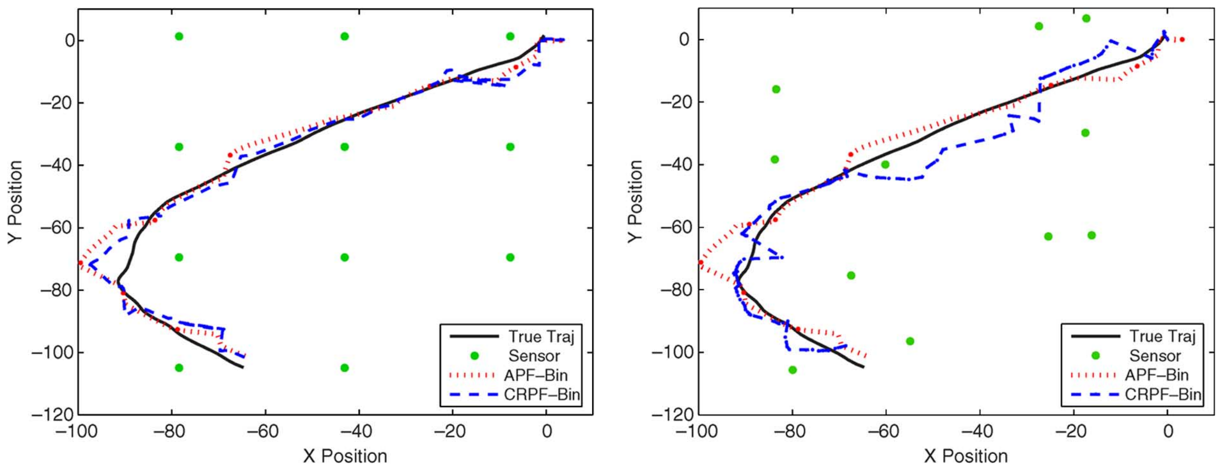


Fig. 2. A realization of a target trajectory and its estimates by APF and CRPF for deterministically and randomly deployed sensor networks.

state noise process was $\mathbf{C}_u = \mathrm{diag}\{0.05, 0.01\}$, and the measurement noise in (2) had mean $\mu_v = 1$ were $\sigma_v^2 = 0.01$, and $\sigma_\epsilon^2 = 0.01$. The noise variance σ_ϵ^2 and the parameters β_n, unless otherwise stated, were chosen to yield signal-to-noise ratio (SNR) of 20 dB at the fusion center. The sampling interval was $T_s = 1$ s.

In the implementation of the APF and CRPF algorithms, we used $M = 1000$ particles. The prior for the target's location and velocity was a Gaussian distribution with mean $\bar{\mathbf{x}}_0 = [0\ 0\ 0.01\ 0.01]^\top$ and covariance matrix $\mathbf{\Xi} = \mathrm{diag}\{10, 10, 0.1, 0.1\}$, and the initial particles of Ψ were drawn from a uniform distribution on $[10^3, 10^4]$. The cost function of the CRPF was defined by (16) and (17) with $\lambda = 0$ and $q = 2$.

We experimented with two sensor networks, with deterministically and randomly deployed sensors. In Fig. 2, we can see the two networks (the sensors are displayed with small circles), a realization of a trajectory of one object and the obtained estimates by APF and CRPF, denoted by APF-bin, and CRPF-Bin, respectively. It can be seen that the algorithms track the target's trajectory closely.

In Fig. 3, we display the root-mean-square errors (RMSEs) of the location estimate of the target obtained by the APF and CRPF algorithms that use complete sensor measurements (APF-Comp and CRPF-Comp, respectively), and the APF and CRPF algorithms based on binary measurements (APF-Bin and CRPF-Bin, respectively) in the deterministic network. The RMSEs in this experiment were obtained by averaging over 100 different realizations. As expected, the performances of APF-Comp and CRPF-Comp were better than the performances of APF-Bin and CRPF-Bin (in RMSE by about 4 m) at any time instant t. It should be observed, too, that the CRPF does not have much degraded performance with respect to the APF even though it does not use probabilistic information. Clearly, the difference in performance depends on various factors including on the number of sensors in the field, their constellation, the actual trajectory of the target, the strength of the transmitted signal by the target, and the sampling interval.

The RMSEs of the location coordinates and velocity components of the target and their PCRBs for the random sensor network are shown in Fig. 4. Again, 100 different realizations were

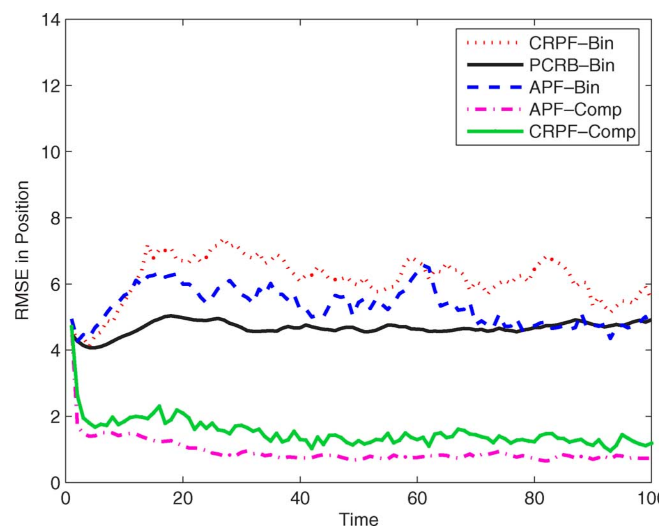


Fig. 3. RMSEs of the location estimates of the target in deterministic network obtained by the APF and CRPF algorithms with complete sensor measurements (APF-Comp and CRPF-Comp, respectively), and the APF and CRPF algorithms with binary measurements (APF-Bin and CRPF-Bin, respectively). The PCRB curve is for binary sensor measurements.

used in the experiment. Similar results were obtained as in the previous experiment. The RMSEs of the constant parameter Ψ are displayed in Fig. 5 for the deterministically deployed sensor network.

In the next set of experiments, we studied the performance of proposed methods with respect to different SNRs at the fusion center. In Fig. 6, we see the performances of these methods expressed by the cumulative distribution functions of the RMSEs of APF-Bin and CRPF-Bin. For example, Fig. 6 shows that the probability of the RMSE being smaller than 6 m is practically one for SNRs greater than or equal to 10 dB. At $\mathrm{SNR} = 5\ \mathrm{dB}$, the probability is almost one if the RMSE is less than or equal to 9 m. From the graphs, we see that the performance of the CRPF-Bin degrades more rapidly than that of the APF-Bin when the SNR decreases.

We further studied the impact of the threshold in detecting the presence of the target through the PCRBs for the deterministic network in the examples. The results shown in Fig. 7 imply that

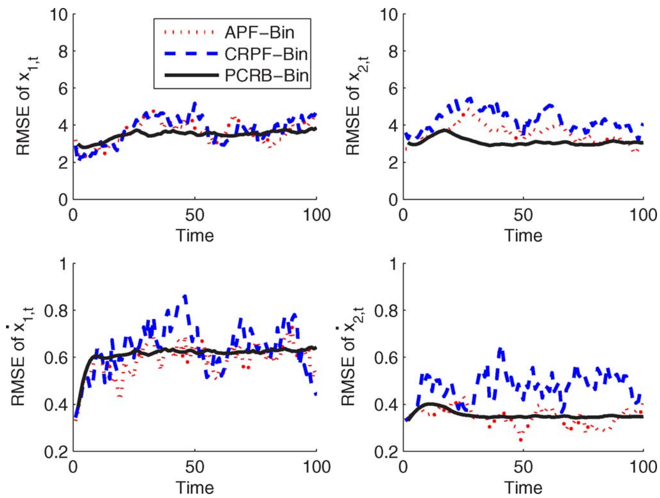


Fig. 4. RMSEs of the locations and velocities obtained by APF-Bin and CRPF-Bin as functions of time obtained in a random network. The respective PCRBs for binary sensor measurements are also plotted.

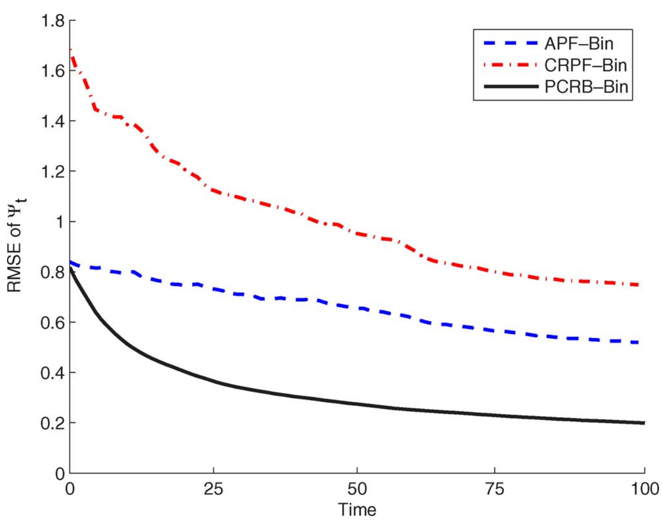


Fig. 5. RMSEs of Ψ as a function of time.

the choice of threshold can be very important for the accuracy of the applied methods. We point out that as the threshold values continue to increase, the PCRBs of the position will start to increase too.

We have also performed simulations when the APF-Bin uses inaccurate distributions of the noise processes in the model. We generated the noise process in the state equations with a mixture $\mathbf{u}_t \sim 0.6\mathcal{N}(\mathbf{0}, \mathbf{C}_{u,1}) + 0.4\mathcal{N}(\mathbf{0}, \mathbf{C}_{u,2})$ with $\mathbf{C}_{u,1} = \mathrm{diag}\{0.05, 0.02\}$ and $\mathbf{C}_{u,2} = \mathrm{diag}\{0.5, 0.2\}$. Instead of the accurate pdf, the APF assumed a Gaussian pdf $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{C}}_u)$ where $\tilde{\mathbf{C}}_u = \mathrm{diag}\{0.01, 0.02\}$. The transmission measurement noise was simulated using a Gaussian mixture as follows:

$$\epsilon_{n,t} \sim 0.5\mathcal{N}\left(\mu_\epsilon, \sigma_\epsilon^2\right) + 0.5\mathcal{N}\left(-\mu_\epsilon, \sigma_\epsilon^2\right)$$

and the APF assumed the measurement noise to be zero mean and with a variance of $\tilde{\sigma}_\epsilon^2 = \mu_\epsilon^2 + \sigma_\epsilon^2$. This ensures that the first two moments of the true and assumed measurement noise statistics are the same. The simulated values of $\tilde{\sigma}_\epsilon^2$ and σ_ϵ^2 were 0.0081 and 0.001, respectively. The RMSEs of 100 different trajectories were computed and summed over the entire time period. In Fig. 8, we plot the cumulative distribution functions of the RMSE of APF-Bin and CRPF-Bin. For example, the graph shows that 95% of the times, the RMSE accumulated by the CRPF-Bin is below 12 m while the RMSE accumulated with the APF-Bin is less than 20 m. Clearly CRPF-Bin outperforms the APF-Bin considerably and this is an important result. In practical applications, frequently the assumed distributions and their parameters may be inaccurate, which may cause the degradation of the APF-Bin. As expected in such scenarios CRPF-Bin performs much more robustly.

VIII. Conclusion

In this paper we focused on the use of binary wireless sensor networks for tracking a single target. This is a rather challenging problem because complete measurements are compressed to binary decisions of whether a target is or is not detected by a sensor. We applied two particle filtering algorithms for processing of the binary data, auxiliary particle filtering and cost-reference particle filtering. The adopted model of sensor measurements was the signal strength, although any other type of measurement model would be equally applicable. We also derived PCRBs of the estimated unknowns. We conducted several sets of experiments, and the simulation results show that the proposed methods track with good accuracy. By no means is the study in this paper complete. There are several interesting topics that deserve full attention including tracking in networks with uncertain sensor locations, tracking in networks with failing sensors, tracking with biased sensors, optimality of deployment of sensors, and choice of sensor thresholds.

Appendix

Here we present the derivation of the PCRBs of tracking a single target. Recall that the dynamics of the state evolution are given by (1), where the prior distribution $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is singular. Let $\boldsymbol{\xi}_t = [\boldsymbol{v}_t \ \tilde{\mathbf{l}}_t]^\top$, where $\tilde{\mathbf{l}}_t = [x_{1,t} \ x_{2,t} \ \Psi_t]^\top$ and $\boldsymbol{v}_t = [\dot{x}_{1,t} \ \dot{x}_{2,t}]^\top$. In [27], a recursive method for determining the PCRBs for such cases is presented, and here we follow that approach. The state evolution as given by (1) and (22) can also be expressed in block vector notation as

$$\begin{aligned} \boldsymbol{v}_t &= \boldsymbol{v}_{t-1} + \mathbf{F}\mathbf{u}_t \\ \tilde{\mathbf{l}}_t &= \tilde{\mathbf{l}}_{t-1} + \mathbf{G}(\boldsymbol{v}_t + \boldsymbol{v}_{t-1}) \end{aligned} \tag{26}$$

where

$$\mathbf{F} = \begin{bmatrix} T_s & 0 \\ 0 & T_s \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \frac{T_s}{2} & 0 \\ 0 & \frac{T_s}{2} \\ 0 & 0 \end{bmatrix}.$$

Let the information matrix of $\boldsymbol{\xi}_t$ be denoted by $\mathbf{J}_t$ (which in our case is of size 5×5). The recursive computation of the PCRBs can then be written as follows [27]:

$$\begin{aligned} \mathbf{J}_t &= \begin{bmatrix} \mathbf{J}_t^{11} & \mathbf{J}_t^{12} \\ \mathbf{J}_t^{21} & \mathbf{J}_t^{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_t^{22} & \mathbf{S}_t^{23} \\ \mathbf{S}_t^{32} & \mathbf{S}_t^{33} \end{bmatrix} - \begin{bmatrix} \mathbf{S}_t^{21} \\ \mathbf{S}_t^{31} \end{bmatrix} \left[\mathbf{S}_t^{11}\right]^{-1} \left[\mathbf{S}_t^{12} \quad \mathbf{S}_t^{13}\right] \end{aligned} \tag{27}$$

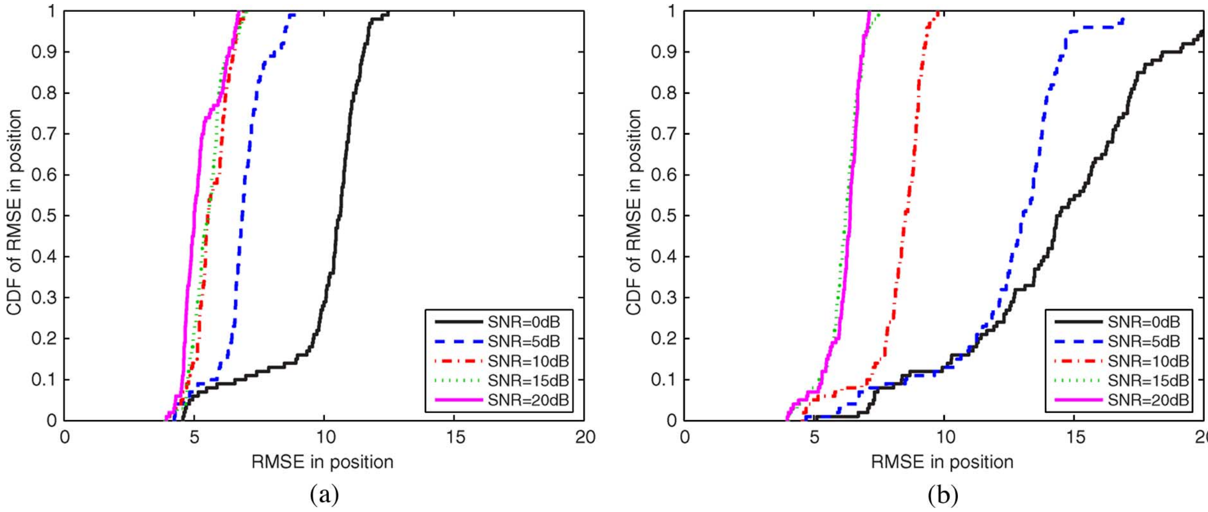


Fig. 6. Performance of the APF-Bin and CRPF-Bin algorithms for various SNRs measured by the cumulative distribution functions of the RMSEs. (a) APF. (b) CRPF.

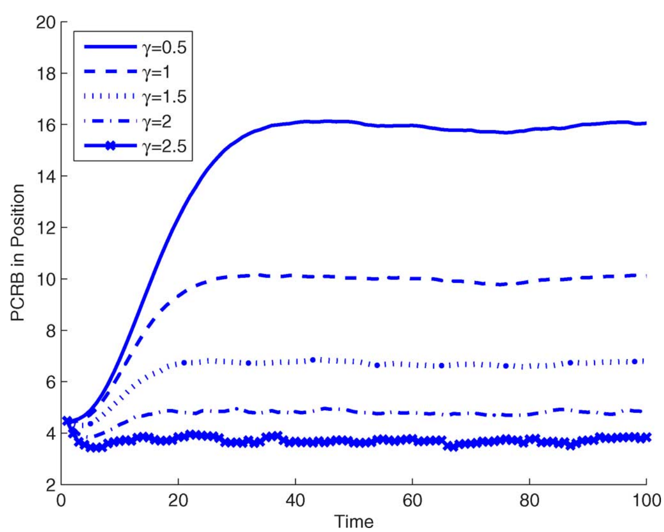


Fig. 7. PCRBs of estimated positions from binary sensor measurements for various sensor thresholds.

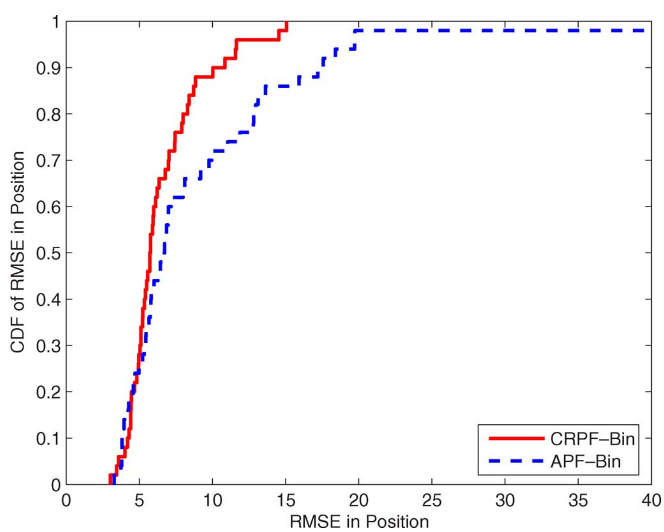


Fig. 8. Cumulative distribution function of the RMSEs of APF-Bin and CRPF-Bin.

where the block matrices have sizes 2×2 ($\mathbf{J}_t^{11}$, $\mathbf{S}_t^{11}$, $\mathbf{S}_t^{13}$, $\mathbf{S}_t^{31}$, and $\mathbf{S}_t^{33}$), 2×3 ($\mathbf{J}_t^{12}$, $\mathbf{S}_t^{12}$ and $\mathbf{S}_t^{32}$), 3×2 ($\mathbf{J}_t^{21}$, $\mathbf{S}_t^{21}$ and $\mathbf{S}_t^{23}$), and 3×3 ($\mathbf{J}_t^{22}$ and $\mathbf{S}_t^{22}$), and the block matrices $\mathbf{S}_t^{ij}$ are computed according to

$$\begin{aligned}\mathbf{S}_t &= \begin{bmatrix} \mathbf{S}_t^{11} & \mathbf{S}_t^{12} & \mathbf{S}_t^{13} \\ \mathbf{S}_t^{21} & \mathbf{S}_t^{22} & \mathbf{S}_t^{23} \\ \mathbf{S}_t^{31} & \mathbf{S}_t^{32} & \mathbf{S}_t^{33} \end{bmatrix} \\ &= \mathbf{M}^{-\top} \begin{bmatrix} \mathbf{J}_{t-1}^{11}+\mathbf{H}_{t-1}^{11} & \mathbf{J}_{t-1}^{12}+\mathbf{H}_{t-1}^{12} & \mathbf{H}_{t-1}^{13} \\ \left(\mathbf{J}_{t-1}^{12}+\mathbf{H}_{t-1}^{12}\right)^\top & \mathbf{J}_{t-1}^{22}+\mathbf{H}_{t-1}^{22} & \mathbf{H}_{t-1}^{23} \\ \left(\mathbf{H}_{t-1}^{13}\right)^\top & \left(\mathbf{H}_{t-1}^{23}\right)^\top & \mathbf{H}_{t-1}^{33} \end{bmatrix} \mathbf{M}^{-1}.\end{aligned}$$

The 7×7 matrix $\mathbf{M}$, and the matrices $\mathbf{H}_t^{ij}$ are defined by

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 3} & \mathbf{I}_{2\times 2} \\ \mathbf{G} & \mathbf{I}_{3\times 3} & \mathbf{G} \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{H}_t^{11} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_t} \log \bar{p}_t\right], \quad \mathbf{H}_t^{12} = E\left[-\Delta_{\boldsymbol{v}_t}^{\tilde{\mathbf{l}}_t} \log \bar{p}_t\right] \\ \mathbf{H}_t^{13} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right], \quad \mathbf{H}_t^{22} = E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\tilde{\mathbf{l}}_t} \log \bar{p}_t\right] \\ \mathbf{H}_t^{23} &= E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right], \quad \mathbf{H}_t^{33} = E\left[-\Delta_{\boldsymbol{v}_{t+1}}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right] \end{aligned}$$

with

$$\bar{p}_t = p(\boldsymbol{v}_{t+1}|\boldsymbol{v}_t)p(\mathbf{z}_{t+1}|\boldsymbol{\xi}_t, \boldsymbol{v}_{t+1}) \tag{28}$$

and Δ being the Laplacian operator.

The computation of the $\mathbf{H}_t^{ij}$, $i, j = 1, 2, 3$ proceeds as follows:

1) $\mathbf{H}_t^{11}$: We use the following identities:

$$\begin{aligned} \mathbf{H}_t^{11} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_t} \log \bar{p}_t\right] = \mathbf{H}_{t,a}^{11} + \mathbf{H}_{t,b}^{11} \\ \mathbf{H}_{t,a}^{11} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_t} \log p(\boldsymbol{v}_{t+1}|\boldsymbol{v}_t)\right] \\ \mathbf{H}_{t,b}^{11} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_t} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right] \end{aligned}$$

where the expectation is over $\{\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1}, \mathbf{z}_{t+1}\}$.

2) $\mathbf{H}_t^{12}$: For computing $\mathbf{H}_t^{12}$, we need the following expressions:

$$\mathbf{H}_t^{12} = E\left[-\Delta_{\boldsymbol{v}_t}^{\tilde{\mathbf{l}}_t} \log \bar{p}_t\right] = \mathbf{H}_{t,a}^{12} + \mathbf{H}_{t,b}^{12}$$
$$\mathbf{H}_{t,a}^{12} = E\left[-\Delta_{\boldsymbol{v}_t}^{\tilde{\mathbf{l}}_t} \log p(\boldsymbol{v}_{t+1}|\boldsymbol{v}_t)\right] = \mathbf{0}$$
$$\mathbf{H}_{t,b}^{12} = E\left[-\Delta_{\boldsymbol{v}_t}^{\tilde{\mathbf{l}}_t} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right]$$

3) $\mathbf{H}_t^{13}$: We can show that

$$\begin{aligned}\mathbf{H}_t^{13} &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right] \\ &= E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_{t+1}} \log p(\boldsymbol{v}_{t+1}|\boldsymbol{v}_t)\right] \\ &\quad + E\left[-\Delta_{\boldsymbol{v}_t}^{\boldsymbol{v}_{t+1}} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right]. \end{aligned} \tag{29}$$

4) $\mathbf{H}_t^{22}$: For computing $\mathbf{H}_t^{22}$, we use

$$\begin{aligned}\mathbf{H}_t^{22} &= E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\tilde{\mathbf{l}}_t} \log \bar{p}_t\right] \\ &= E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\tilde{\mathbf{l}}_t} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right].\end{aligned}$$

5) $\mathbf{H}_t^{23}$: The process is similar as before. We have

$$\begin{aligned}\mathbf{H}_t^{23} &= E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right] \\ &= E\left[-\Delta_{\tilde{\mathbf{l}}_t}^{\boldsymbol{v}_{t+1}} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right].\end{aligned}$$

6) $\mathbf{H}_t^{33}$: We use the identities

$$\begin{aligned}\mathbf{H}_t^{33} &= E\left[-\Delta_{\boldsymbol{v}_{t+1}}^{\boldsymbol{v}_{t+1}} \log \bar{p}_t\right] \\ &= E\left[-\Delta_{\boldsymbol{v}_{t+1}}^{\boldsymbol{v}_{t+1}} \log p(\boldsymbol{v}_{t+1}|\boldsymbol{v}_t)\right] \\ &\quad + E\left[-\Delta_{\boldsymbol{v}_{t+1}}^{\boldsymbol{v}_{t+1}} \log p(\mathbf{z}_{t+1}|\boldsymbol{v}_t, \tilde{\mathbf{l}}_t, \boldsymbol{v}_{t+1})\right].\end{aligned}$$

References

[1] A. Arora *et al.*, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Comput. Netw.*, pp. 605–634, 2004.

[2] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, and D. Rus, "Tracking a moving object with a binary sensor network," in *Proc. 1st Int. Conf. Embedded Networked Sensor Syst.*, Los Angeles, CA, 2003, pp. 150–161.

[3] M. F. Bugallo, J. Míguez, and P. M. Djurić, "Positioning by cost reference particle filters: Study of various implementations," presented at the 2005 Int. Conf. "Computer as a tool," EUROCON, Belgrade, Serbia and Montenegro, 2005.

[4] J.-F. Chamberland and V. V. Veeravali, "Decentralized detection in sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 407–416, Feb. 2003.

[5] B. Chen, W. B. Heinzelman, M. Liu, and A. T. Campbell, "Editorial of special issue on wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2005, no. 4, pp. 459–461, 2005.

[6] C.-Y. Chong and S. P. Kumar, "Sensor networks: Evolution opportunities and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[7] N. Chopin, "A sequential particle filter for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002.

[8] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 736–746, Mar. 2002.

[9] P. M. Djurić, M. F. Bugallo, and J. Míguez, "Density assisted particle filters for state and parameter estimation," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process., Montreal, QC, Canada, 2004.

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

[11] F. Gustaffson, F. Gunnarsson, N. Bergman, U. Forssel, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filtering for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[12] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham, "On target tracking with binary proximity sensors," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw., (IPSN)*, 2005, pp. 301–308.

[13] J. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[14] S. Kumar, F. Zhao, and D. Shepherd, "Collaborative signal and information processing in microsensor networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 13–14, Mar. 2002.

[15] K. Mechitov, S. Sundresh, Y. Kwon, and G. Agha, "Cooperative tracking with binary-detection sensor networks," in *Proc. 1st ACM Int. Conf. Embedded Networked Sensor Syst.*, 2003, pp. 332–333.

[16] J. Míguez, M. F. Bugallo, and P. M. Djurić, "A new class of particle filters for random dynamical systems with unknown statistics," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 15, pp. 2278–2294, 2004.

[17] J. Míguez, M. F. Bugallo, and P. M. Djurić, "Erratum on a new class of particle filters for random dynamical systems with unknown statistics," *EURASIP J. Appl. Signal Process.*, 2006, Article ID 78708.

[18] J. Míguez and P. M. Djurić, "Blind equalization of frequency selective channels by sequential importance sampling," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2738–2748, Oct. 2004.

[19] P. Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems With Applications*. New York: Springer-Verlag, 2004.

[20] R. Niu and P. Varshney, "Target location estimation in wireless sensor networks using binary data," presented at the 38th Annu. Conf. Inf. Sci. Syst., Princeton, NJ, 2004.

[21] S. Oh and S. Sastry, "Tracking on a graph," in *Proc. 4th Int. Symp. Inf. Process. Sensor Networks (IPSN)*, Los Angeles, CA, 2005, pp. 195–202.

[22] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 5, pp. 2137–2148, May 2003.

[23] M. Pitt and N. Shepard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Statist. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.

[24] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds., *Wireless Sensor Networks*. New York: Springer, 2004.

[25] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan. 2005.

[26] G. Storvik, "Particle filters for state-space models with presence of unknown static parameters," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 281–289, Feb. 2002.

[27] P. Tichavský, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[28] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968.

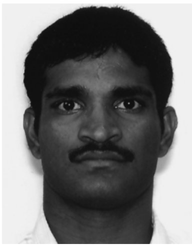
[29] L. Y. Wang, J.-F. Zhang, and G. G. Yin, "System identification using binary sensors," *IEEE Trans. Autom. Control*, vol. 48, no. 11, pp. 1892–1907, Nov. 2003.

[30] F. Zhao and L. Guibas, *Wireless Sensor Networks*. New York: Morgan Kaufmann, 2004.

Petar M. Djurić (F'06) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, in 1981 and 1986, respectively, and the Ph.D. degree in electrical engineering from the University of Rhode Island in 1990.

From 1981 to 1986, he was a Research Associate with the Institute of Nuclear Sciences, Vinča, Belgrade. Since 1990, he has been with Stony Brook University, where he is a Professor in the Department of Electrical and Computer Engineering. He works in the area of statistical signal processing, and his primary interests are in the theory of modeling, detection, estimation, and time series analysis and its application to a wide variety of disciplines.

Prof. Djurić has served on numerous technical committees for the IEEE and has been invited to lecture at universities in the United States and overseas. He is the current Vice President-Finance of the IEEE Signal Processing Society and is Distinguished Lecturer of the Society for the period 2008–2009. He has been an Associate Editor and has served on the Editorial Boards of several journals/magazines in the areas of signal processing and communications.



Mahesh Vemula (S'00) received the B.S. degree from Institute of Technology, Banaras Hindu University (IT-BHU), in 2001 and the Ph.D. degree in electrical engineering from Stony Brook University, NY, in 2007. His thesis on Monte Carlo methods for signal processing applications in sensor networks was supervised by P. M. Djuric.

He is currently a quantitative analyst at a financial firm in Chicago, IL. His research interests include financial engineering, statistical signal processing, Bayesian spectral estimation, localization, and tracking in sensor networks. He has been an intern at Advanced Acoustic Concepts and Bell Labs (now Lucent-Alcatel), Crawford Hill, NJ. He was a visiting scholar at the Universidad Carlos III de Madrid, Spain, in the fall 2006.

Mónica F. Bugallo (M'99) received the Ph.D. degree in computer engineering from the University of A Coruña, Spain, in 2001.

From 1998 to 2000, she was with the Departamento de Electrónica y Sistemas, Universidade da Coruña, Spain, where she worked on interference cancellation applied to multiuser communication systems. In 2001, she joined the Department of Electrical and Computer Engineering, Stony Brook University, NY, where she is currently Assistant Professor and teaches courses in digital communications and information theory. Her research interests are in the field of statistical signal processing, with emphasis on Bayesian analysis, sequential Monte Carlo methods, adaptive filtering, stochastic optimization, and their applications to target tracking, sensor networks, biology, and communications.