DECISION FUSION FOR DISTRIBUTED TARGET TRACKING USING COST REFERENCE PARTICLE FILTERING

Joaquín Míguez[†], *Mónica F. Bugallo*[‡], *Petar M. Djurić*[‡]

[†]Depto. Teoría de la Señal y Comunicaciones, Universidad Carlos III
Avenida de la Universidad 30, Leganés, 28911 Madrid, Spain
[‡]Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, New York 11794, USA.
E-mail: jmiguez@ieee.org, {monica,djuric}@ece.sunysb.edu

ABSTRACT

In this paper, we consider the problem of tracking an object which moves along a certain 2-dimensional area monitored by a network of wireless sensors. We propose a novel decision fusion algorithm for target tracking based on a recently proposed sequential Monte Carlo (SMC) methodology called *cost-reference particle filtering* (CRPF). Computer simulations reveal the feasibility of the proposed method.

1. INTRODUCTION

There has been a recent surge of interest in the use of networks of wireless microsensors to perform a variety of signal processing tasks [1, 2, 3]. The key to most applications is the development of effective algorithms for integrating the information provided by the sensors [3]. When raw data are collected for processing at a central node, *data fusion* algorithms are necessary [3, 4, 5]. A more complex problem arises when some processing is performed locally at the sensors, which produce a simple (often binary) decision for transmission to the central node. This setup is practically appealing because it minimizes network communications, but it usually requires complex *decision fusion* methods [3]. Research progress on decision fusion algorithms for target detection and tracking can be found [6] and [3], respectively.

In this paper, we consider the problem of tracking an object that moves along a certain 2-dimensional area monitored by a network of wireless sensors. We assume the sensors can measure some distance-related physical magnitude (e.g., the received signal strength) and use it to make a binary decision regarding the presence of the target within a certain range. The resulting bit (1/0) is transmitted to a central node, where the local decisions are integrated to recursively estimate the current position and speed of the target.

We propose a novel decision fusion algorithm for target tracking based on the sequential Monte Carlo (SMC) paradigm [7]. Specifically, we investigate a recently proposed SMC methodology called *cost-reference particle filtering* (CRPF) [8]. It is built on the basic principle of standard particle filtering, i.e., the exploration of the space of the signal of interest using randomly generated sample trajectories (termed *particles*) which are adequately weighted according to the available observations. However, and unlike conventional particle filters, CRPF techniques do not require an explicit probabilistic model of the involved signals. Instead, they depend on an arbitrary cost function (not necessarily tied to any data statistics) and can attain substantial advantage in terms of simplicity and robustness.

The remaining of this paper is organized as follows. Section 2 introduces the signal model used in the considered problem. The posterior Cramér-Rao bound (PCRB) is derived in Section 3. In Sections 4 and 5, the basics of the CRPF methodology and the specifics of the considered algorithms are discussed, respectively. Computer simulation results that illustrate the performance of the algorithm are presented in Section 6.

2. SIGNAL MODEL

It is of interest to recursively estimate the position, $\mathbf{r}_t = [r_{t,x}, r_{t,y}]^\top$, and the speed, $\mathbf{s}_t = [s_{t,x}, s_{t,y}]^\top$, at discrete-time t (signal samples are obtained with a period of T_s s) of a target moving along the $\mathbb{R}^2$ plane. The time evolution of the 4×1 state vector $\mathbf{x}_t = [\mathbf{r}_t^\top, \mathbf{s}_t^\top]^\top$ is given by the linear kinematic model

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{Q}\mathbf{u}_t, \qquad (1)$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{I}_2 & T_s\mathbf{I}_2 \\ \mathbf{0}_{2\times2} & \mathbf{I}_2 \end{bmatrix}$, T_s is the system observation period, $\mathbf{I}_n$ is the $n \times n$ identity matrix, $\mathbf{Q} = \begin{bmatrix} \frac{T_s^2}{2}\mathbf{I}_2 & \mathbf{0}_{2\times2} \\ \mathbf{0}_{2\times2} & T_s\mathbf{I}_2 \end{bmatrix}$ and $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}_{4\times1}, \mathbf{I}_2)$ is a Gaussian noise vector.

The network consists of N randomly deployed sensors (with uniform distribution and density D_s units/m^2) on the region where the target needs to be tracked. The n-th sensor is located at a known position $\mathbf{r}_n$, and when a target is present, is able to measure some physical magnitude related to the distance between the target and the sensor, denoted as $d_{n,t} = ||\mathbf{r}_t - \mathbf{r}_n||$. A typical example is the received signal power, which can be converted into a distance measurement if a model of the signal propagation attenuation is available. In any case, we assume the resulting measurements are corrupted with noise and yield

$$\tilde{d}_{n,t} = d_{n,t} + l_{n,t}, \quad n = 1, ..., N, \qquad (2)$$

where $l_{n,t}$ is a zero-mean perturbation. In this paper, we assume $l_{n,t} \sim \ell(0, b)$, where $\ell(z|a,b) = \frac{1}{2b}\exp\{-|z-a|/b\}$ is the Laplacian probability density function (pdf). Using

J. Míguez acknowledges the support of Xunta de Galicia (project PGIDIT04TIC105008PR). M. F. Bugallo and P. M. Djurić acknowledge the support of National Science Foundation (Award CCR-0082607).

these distance estimates, the sensor makes a binary decision

$$y_{n,t} = \begin{cases} 1 & \text{if } \tilde{d}_{n,t} < {}_1 \vee \left(\tilde{d}_{n,t} < {}_2 \wedge y_{n,t-1} = 0\right) \\ 0 & \text{if } \tilde{d}_{n,t} > {}_2 \vee \left({}_1 < \tilde{d}_{n,t} \wedge y_{n,t-1} = 1\right) \end{cases}, \quad (3)$$

where $\vee$ and $\wedge$ denote logical `or` and `and` operations, respectively, and $0 < {}_1 < {}_2$ are distance thresholds. At the central node, a decision fusion algorithm is used to estimate the target trajectory, $\mathbf{x}_{0:t} = \{\mathbf{x}_0, \dots, \mathbf{x}_t\}$, from the sequence $\mathbf{y}_{1:t}$, where $\mathbf{y}_k = [y_{1,k}, \dots, y_{N,k}]^\top$ is the $N \times 1$ vector of local decisions.

3. POSTERIOR CRAMÉR-RAO BOUND

The covariance matrix of any estimator $\hat{\mathbf{x}}_t$ computed from the observations $\mathbf{y}_{1:t}$ can be lower bounded using the PCRB

$$\mathrm{Cov}(\hat{\mathbf{x}}_t) - \mathbf{F}_t^{-1} \geq \mathbf{0}, \quad (4)$$

where $\mathbf{F}_t$ is the posterior Fisher information submatrix for $\mathbf{x}_t$ and $\geq \mathbf{0}$ indicates that the matrix on the left-hand side is positive semi-definite. Following [9], $\mathbf{F}_t$ can be recursively computed as

$$\mathbf{F}_t = \mathbf{\Sigma}_u^{-1} - \mathbf{J}_t - \mathbf{\Sigma}_u^{-1}\mathbf{A}\left(\mathbf{F}_{t-1} + \mathbf{A}^\top\mathbf{\Sigma}_u^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^\top\mathbf{\Sigma}_u^{-1}, \quad (5)$$

where $\mathbf{\Sigma}_u = \mathbf{Q}\mathbf{Q}^\top$ is the covariance matrix of the noise term $\mathbf{Q}\mathbf{u}_t$ in (1) and $\mathbf{J}_t = \mathrm{E}_{p(\mathbf{y}_t,\mathbf{x}_t)}\left[\frac{{}^2}{\mathbf{x}_{t,i}\ \ \mathbf{x}_{t,j}} \log p(\mathbf{y}_t|\mathbf{x}_t)\right]$ is the Jacobian matrix of the log-likelihood. It must be noted here, however, that the expectation in the definition of $\mathbf{J}_t$ cannot be found in closed-form and must be estimated by Monte Carlo integration.

The use of a double threshold, ${}_i$, $i = 1, 2$, in the local decision function (3) instead of a single threshold can be justified using the PCRB. Figure 1 shows the curves $PCRB_t = \mathrm{trace}\left(\mathbf{F}_t^{-1}\right)$ (which is the sum of the minimum achievable variances for each component in $\mathbf{x}_t$) for two different systems. It is observed that making the local decisions using a double threshold yields better tracking accuracy than using a single threshold. We note that this is achieved at the expense of a negligible increase in computational complexity and no communication overhead at all.

4. COST REFERENCE PARTICLE FILTERING

A new class of SMC methods, termed cost reference particle filtering (CRPF), has been recently proposed in [10]. The new technique preserves the philosophy of exploring the state space by generating random weighted particles, but differs substantially from standard particle filtering algorithms. We summarize the new methodology in this section.

Instead of imposing an explicit probabilistic model on the system (1)-(3), let us assume the availability of a real, lower-bounded and additive cost function of the form

$$\mathscr{C}(\mathbf{x}_{0:t}, \mathbf{y}_{1:t},\ \ ) \triangleq \ \ \mathscr{C}(\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1},\ \ ) + \triangle\mathscr{C}(\mathbf{x}_t, \mathbf{y}_t) \quad (6)$$

where the forgetting factor $\ \ < 1$ avoids attributing an excessive weight to old observations. The cost yields a quantitative assessment of the state sequence, $\mathbf{x}_{0:t}$, in view of the observations, $\mathbf{y}_{1:t}$. We also introduce a one-step *risk* function $\mathscr{R}(\mathbf{x}_{t-1}, \mathbf{y}_t)$ that measures the adequacy of the state at time $t-1$ given the new observation, $\mathbf{y}_t$. A natural choice of the risk (for zero mean state noise) is $\mathscr{R}(\mathbf{x}_{t-1}, \mathbf{y}_t) = \triangle\mathscr{C}(\mathbf{A}\mathbf{x}_{t-1}, \mathbf{y}_t)$.

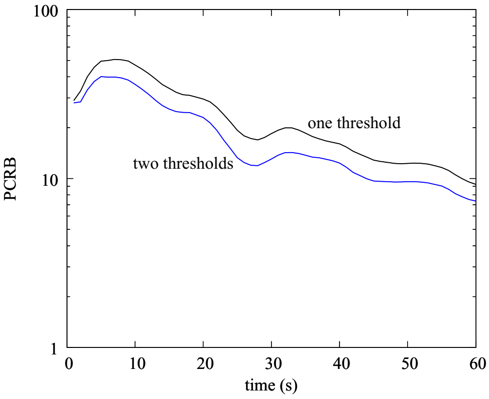

Figure 1: Comparison of the PCRB for systems with single-threshold and double-threshold decisions. Curves obtained by averaging 2000 independent simulations. $T_s = 0.25$ s, $D_s = 10^{-2}$ sensors/m^2,

Given the set $\ \ _t = \{\mathbf{x}_{0:t}^{(m)}, \mathscr{C}_t^{(m)}\}_{m=1}^M$, where $\mathscr{C}_t^{(m)} \triangleq \mathscr{C}(\mathbf{x}_{0:t}^{(m)}, \mathbf{y}_{1:t},\ \ )$, the CRPF algorithm proceeds sequentially as follows:

1. **Initialization.** Assuming $\mathbf{x}_0 \in I_0 \subset \mathbb{R}^{n_x}$, sample $\{\mathbf{x}_0^{(m)}\}_{m=1}^M$ using an arbitrary pdf. The costs are set to a single constant, $\mathscr{C}_0^{(m)} \equiv cst$, $m = 1, \dots, M$.
2. **Recursive loop.** At time $t+1$, perform:
   (a) **Selection.** The most promising particles up to time t are stochastically selected for propagation according to their risk,

   $$\mathscr{R}_{t+1}^{(m)} \triangleq \ \ \mathscr{C}_t^{(m)} + \mathscr{R}(\mathbf{x}_t^{(m)}, \mathbf{y}_{t+1}).$$

   The selection yields an intermediate particle set, $\hat{\ }_t = \left\{\hat{\mathbf{x}}_{0:t}^{(m)}, \hat{\mathscr{C}}_t^{(m)}\right\}_{m=1}^M$.

   (b) **Propagation.** For $m = 1, \dots, M$, let

   $$\begin{array}{rcl} \mathbf{x}_{t+1}^{(m)} & \sim & p_{t+1}(\mathbf{x}|\hat{\mathbf{x}}_t^{(m)}) \\ \mathscr{C}_{t+1}^{(m)} & = & \ \ \hat{\mathscr{C}}_t^{(m)} + \triangle\mathscr{C}_{t+1}^{(m)} \end{array}$$

   where $\triangle\mathscr{C}_{t+1}^{(m)} \triangleq \triangle\mathscr{C}_{t+1}(\mathbf{x}_{t+1}^{(m)}, \mathbf{y}_t)$, and p_{t+1} is an arbitrary propagation pdf.
3. **Estimation.** We suggest to use the probability mass function (pmf)

   $$\ {}_t^{(m)} \quad\quad (\mathscr{C}_t^{(m)})$$

   to compute

   $$\mathbf{x}_t^{mean} = \overset{M}{\underset{m=1}{}} \ \mathbf{x}_t^{(m)} \ \ {}_t^{(m)},$$

   which attains an asymptotically minimum cost when function $\ \ $ is adequately chosen. In principle, we only require that $\ \ $ be monotonically decreasing and will refer to it as a *generating function* in the sequel. It should be noted that, although it is used to define the pmf $\ \ $, $\ \ $ is

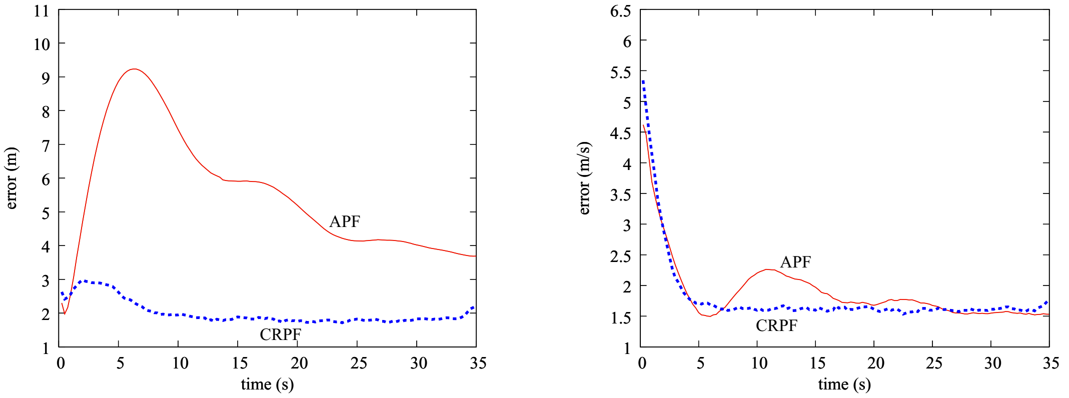

Figure 2: Comparison of standard APF and CRPF algorithm. **Left:** Deviation from the true trajectory (m) when $M = 50$ particles. **Right:** Deviation from the true speed (m/s) when $M = 50$ particles.

not a pmf itself (it is defined on a continuous space), and not necessarily a pdf (although some pdf's can be used as generating functions).

For comparison with the standard particle filters we consider the CRPF with forgetting factor $\ = 0$, where the costs depend only on the particles at time t, i.e., $\mathcal{C}_t^{(m)} = \triangle\mathcal{C}_t^{(m)} = \triangle\mathcal{C}(\mathbf{x}_t^{(m)}, \mathbf{y}_t)\ \forall m \in \{1, ..., M\}$. We will restrict our attention to CRPF algorithms where the selection is performed at each time step using standard resampling methods, although less restrictive possibilities exist (e.g., the local selection method proposed in [10] or the proposal of particles that minimizes the risk).

Different combinations of the cost function, $\mathcal{C}$, the particle propagation pdf, p_t, and the *generating* function yield different algorithms. In particular, it can be interpreted that each combination of and $\mathcal{C}$ yields an implicit probabilistic model for the system, since they allow the assignment of probability masses to the particles. Compared to standard SMC techniques, CRPF algorithms are more flexible, more robust, and in general, easier to implement.

5. ALGORITHMS

We have applied the standard auxiliary particle filter (APF) [11] and the CRPF-type algorithm to the tracking of $\mathbf{x}_{0:t}$ from $\mathbf{y}_{1:t}$.

The APF algorithm samples from the prior pdf, which has the mixture Gaussian form

$$q_M(\mathbf{x}_t) = \sum_{m=1}^{M} w_{t-1}^{(m)} \mathcal{N}(\mathbf{x}_t | \mathbf{A}\mathbf{x}_{t-1}^{(m)}, \mathbf{Q}\mathbf{Q}^\top), \tag{7}$$

and use the correct form of the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ to compute the new weights. Assuming the noise processes $l_{n,t}$, $n = 1, ..., N$, are independent and identically distributed $l_{n,t} \sim \ell(0, b)$ and taking into consideration that the measurements $y_{n,t}$, $n = 1, ..., N$, are computed according to (3), it is straightforward to find that

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{x}_t) \quad = \quad & \prod_{n=1}^{N} [\ \ (y_{n,t-1})\mathcal{L}(\ \ _2|d_{n,t}, b) + \\ & (1 - y_{n,t-1})\mathcal{L}(\ \ _1|d_{n,t}, b)], \end{aligned} \tag{8}$$

where

$$\mathcal{L}(\ \ |a, b) = \int_{-} \ell(z|a, b)dz$$

is the Laplacian distribution function with parameters a, b and $\ (\cdot)$ is the Kronecker's delta function. The algorithm is summarized in Table 1.

```
Given $_t = \{\mathbf{x}_t^{(m)}, w_t^{(m)}\}_{m=1}^{M}$, perform:
    $q_{t+1}^{(i)} \quad w_t^{(i)} p(\mathbf{y}_{t+1}|\mathbf{A}\mathbf{x}\ _t^{(i)})$        $i = 1, ..., M$
    $k^{(m)} \sim p(k) = q_{t+1}^{(i)}$                     $m = 1, ..., M$
    $\mathbf{x}_{t+1}^{(m)} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(k^{(m)})}) = \mathcal{N}(\mathbf{A}\mathbf{x}\ _t^{(k^{(m)})}, \mathbf{Q}\mathbf{Q}\ ^\top)$
    $w_{t+1}^{(m)} = w_t^{(m)} \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}^{(m)})}{p(\mathbf{y}_{t+1}|\mathbf{A}\mathbf{x}_t^{(k^{(m)})}})$
    Normalize weights
```

Table 1: Recursive step of the APF algorithm.

In order to define the cost function for the CRPF, let $d_n(\mathbf{x}_t) = ||\mathbf{r}_t - \mathbf{r}_n||$ and $r > 0$ be an arbitrary radius, and define $y_n(\mathbf{x}_t, r)$ according to eq. (9). If we let $\mathbf{y}(\mathbf{x}'_t) = \lim_{r \to 0^+} \mathbf{y}(\mathbf{x}'_t, r)$ and note that $\mathbf{y}(\mathbf{x}'_t) \in \{1, 0\}^N$, we can build a simple cost function using the Hamming distance (denoted as $h(\cdot, \cdot)$),

$$\triangle\mathcal{C}(\mathbf{x}'_t, \mathbf{y}_t) = h\left(\mathbf{y}_t, \mathbf{y}(\mathbf{x}'_t, r)\right). \tag{10}$$

We build a generating function according to

$$\ (\mathcal{C}_t) = \ \left(\left|\mathcal{C}_t - \min_k \left\{\mathcal{C}_t^{(k)}\right\}\right|\right). \tag{11}$$

where $\ $ is a monotonically decreasing function defined by

$$\ (z) = z^{-} \ \exp\{- \ z\}, \quad z \geq 0, \tag{12}$$

with and being positive constants.

6. COMPUTER SIMULATIONS

We have carried out computer simulations to illustrate the performance of the considered algorithms. The model parameters for the numerical experiments are as follows: $T_s = 0.25$ s, $D_s = 10^{-2}$ sensors/m^2, $\ _1 = 14$ m, $\ _2 = 16$ m and $b = 1/\sqrt{2}$ (unit variance of the measurement Laplacian noise). The starting target position and speed are randomly drawn from $\mathcal{N}(\mathbf{0}, 10\mathbf{I}_4)$ and the CRPF algorithm with continuous cost function employ $r = 4$ m as a radius.

We have studied both the percentage of correct tracks attained by each algorithm and the accuracy of estimation. A 'correct track' is achieved when the distance between the

$$y_n(\mathbf{x}_t, r) = \begin{cases} 1 & \text{if } (d_n(\mathbf{x}_t) < {}_1 - r) \vee [(d_n(\mathbf{x}_t) < {}_2 - r) \wedge (y_{n,t-1} = 0)] \\ 0 & \text{if } [(d_n(\mathbf{x}_t) > {}_1 + r) \wedge (y_{n,t-1} = 1)] \vee (d_n(\mathbf{x}_t) > {}_2 + r) \\ \frac{-1}{2r} d_n(\mathbf{x}_t) + \frac{{}_1 + r}{2r} & \text{if } ({}_1 - r < d_n(\mathbf{x}_t) < {}_1 + r) \wedge (y_{n,t-1} = 1) \\ \frac{-1}{2r} d_n(\mathbf{x}_t) + \frac{{}_2 + r}{2r} & \text{if } ({}_2 - r < d_n(\mathbf{x}_t) < {}_2 + r) \wedge (y_{n,t-1} = 0) \end{cases} \quad (9)$$

true target position, $\mathbf{r}_t$, and the position estimate provided by the particle filtering algorithm, $\hat{\mathbf{r}}_t$, after 35 s is $||\mathbf{r}_t - \hat{\mathbf{r}}_t|| < \frac{1}{\sqrt{D_s}}$ (i.e., we have uncertainty only up to the coverage area assigned to a single sensor). Figure 2 shows the accuracy of estimation averaged over 1000 independent simulation runs by means of the position mismatch (left), i.e., given the true position and its estimate, we compute the error signal $e_{p_t} = ||\mathbf{r}_t - \hat{\mathbf{r}}_t||$ (m) for each algorithm and each simulation run, and the speed mismatch (right), i.e., given the true speed, $\mathbf{v}_t$, and its estimate, $\hat{\mathbf{v}}_t$, we compute the error signal $e_{v_t} = ||\mathbf{v}_t - \hat{\mathbf{v}}_t||$ (m/s) for each algorithm and each simulation run.

Finally, Figure 3 depicts the mean square error (MSE) computed for the last 5 s of simulation for different number of particles, $MSE = {}_{t=\lfloor \frac{30}{T_s} \rfloor}^{\lfloor \frac{35}{T_s} \rfloor} ||\mathbf{x}_t - \hat{\mathbf{x}}_t||^2$, where $\hat{\mathbf{x}}_t$ is the full state estimate. In this case 3000 runs have been averaged to obtain the plots (1000 for each value of the number of particles, M).

In summary, our numerical results indicate that the CRPF algorithm is more reliable (higher number of correct tracks) and accurate when the number of particles, M, is low (this is coherent with the numerical study of a navigation problem in [10]). As M grows, the algorithms become almost equally reliable. Note that the CRPF algorithm is not designed to optimize the MSE between the true state sequence and its estimate, hence we cannot expect any 'optimal' performance in this sense. In exchange, it is insensitive to the statistics of the observations, which need to be known in detail in order to design the APF.

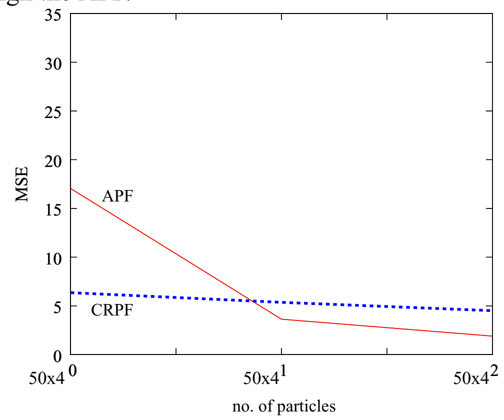

Figure 3: MSE comparison of standard APF and CRPF algorithm for correct tracks when $M = 50$, $M = 200$ and $M = 800$.

7. CONCLUSIONS

We have proposed a new method to solve the problem of tracking an object which moves along a certain 2-dimensional area monitored by a network of wireless sensors that transmit binary decisions to a fusion center. The fusion algorithm is based on the recently proposed cost-reference particle filtering (CRPF) methodology which substitutes the explicit probabilistic model required in classical particle filtering methods by an arbitrary cost function (not necessarily tied to any data statistics). The resulting method attains substantial advantage in terms of simplicity and robustness compared to those based on the traditional approach.

REFERENCES

[1] R. C. Luo, C.-C. Yih, and K. L. Su, "Multisensor fusion and integration: approaches, applications and future research directions," *IEEE Sensors Journal*, vol. 2, no. 2, pp. 107–119, April 2002.

[2] C.-Y. Chong and S. C. Kumar, "Sensor networks: Evolution, opportunities and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[3] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "Distributed target classification and tracking in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163–1171, August 2003.

[4] R. Lobbia and M. Kent, "Data fusion of decentralized local tracker outputs," *IEEE Transactions Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 787–799, July 1994.

[5] Q. Gan and C. J. Harris, "Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion," *IEEE Transactions Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 273–280, January 2001.

[6] R. S. Blum, S. A. Kassam, and H. V. Poor, "Distributed detection with multiple sensors: Part ii – advanced topics," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 64–79, January 1997.

[7] M. S. Aralumpalam, S. Maskell, N. Gordon, and T. Klapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[8] J. Míguez, S. Xu, M. F. Bugallo, and P. M. Djurić, "Particle filtering for systems with unknown probability distributions," in *Proceedings of the IEEE Workshop on SSP*, September 2003.

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

[10] 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, November 2004.

[11] M. K. Pitt and N. Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 13, pp. 273–293. Springer, 2001.