TARGET TRACKING WITH ASYNCHRONOUS MEASUREMENTS BY A NETWORK OF DISTRIBUTED MOBILE AGENTS

Jonathan Beaudeau, Mónica F. Bugallo and Petar M. Djurić

Department of Electrical & Computer Engineering
Stony Brook University, Stony Brook, NY 11794-2350 (USA)
email: JonBeaudeau@aol.com, monica@ece.sunysb.edu, djuric@ece.sunysb.edu

ABSTRACT

In this paper we consider the problem of target tracking in a network of mobile agents that receive asynchronous measurements. The agents measure received signal strengths from the target and broadcast the information to the remaining agents engaged in the tracking. We propose several non-centralized schemes based on particle filtering that account for the lack of synchronization. We demonstrate the proposed methods by computer simulations and compare their performance to the synchronous scenario. The obtained results reveal that the proposed strategies efficiently compensate for the asynchronism of the measurements.

Index Terms— Particle filtering, target tracking, mobile agents, asynchronous measurements.

1. INTRODUCTION

New advances in technology have made the concept of target tracking using mobile sensor networks a reality [1]. Development of a comprehensive framework for such an environment is not a trivial problem, and recently it has commanded significant focus within the research community. Numerous issues related to sensor motion control, communication, measurement synchronization, and localization need to be resolved before a feasible implementation can be provided.

Most of existing work on mobile sensor networks [2, 3] has focused on solutions for scenarios where the agents communicate their measurements to a central unit. This unit is in charge of processing the data and issuing appropriate motion control commands. Tracking is carried out by particle filtering [4] or cost-reference particle filtering [5]. In our previous work [6], we proposed a decentralized solution and addressed the problem of maintaining proper tracking in presence of stationary interfering sources.

In this paper we address the problem of tracking in a network of mobile agents that receive asynchronous measurements. Several different algorithms are proposed which attempt to restore the performance degradation caused by the loss of synchronism. Related work has been reported in [7], which introduces a reformulation of the original target estimation problem in a more suitable form of the asynchronous measurements. There, the agents are static and the measurements are binary signals. Here we utilize a more complex measurement model based on received signal strength, consider decentralize tracking based on particle filtering and maintain mobility of the agents.

This work was supported by NSF under Awards CCF-0953316 and CCF-1018323 and by the ONR under Award N00014-09-1-1154.

2. PROBLEM FORMULATION

A certain area is under surveillance by mobile agents. When they are idle, they are static and on alert to detect a target that enters the area and track it. The agents are equipped with sensors that acquire signals from the target, which are used for tracking. Once the targets leave the area of surveillance, the agents stop tracking the target and possibly hand it off to a group of agents that are tasked to monitor a neighboring area. We consider the situation where agents receive signals from the target in an asynchronous way. For ease of modeling, we introduce the concept of an agent's *base time* defined as the instant at which it receives a signal from the target. For a given absolute interval of time from $t-1$ to t, all agents receive signals from the target at their own base time and broadcast their measurements to one another. Additionally, we impose the constraint that agents estimate (at a minimum) the target's location at their own base time. We introduce a new notation that defines each agent's base time; within the interval $t-1$ to t, where $t = T, 2T, \cdots$, the i-th agent receives the signal at time instant $(t-1)+\tau_i$. For simplicity, we order the sensor "measurement instances" according to $\tau_{i-1} < \tau_i$. Variables which take on a value at this time instant will have a subscript (t,i), referring to the corresponding base time for the i-th agent.

The target dynamics and the nature of the state noise will be first described in terms of absolute time. Then, we will relate this model to the new notation just described. In particular, we assume that a target moves in a 2-D plane according to the Markovian model

$$\boldsymbol{x}_{t_2} \quad = \quad \mathbf{A}(t_2,t_1)\boldsymbol{x}_{t_1} + \boldsymbol{u}_{t_2,t_1}, \qquad (1)$$

where $\boldsymbol{x}_{t_2}$ is a state vector defined by $\boldsymbol{x}_{t_2} = [x_{1,t_2}\ x_{2,t_2}\ \dot{x}_{1,t_2}\ \dot{x}_{2,t_2}]^\top$, with x_{1,t_2} and x_{2,t_2} being the coordinates of the target in the two-dimensional Cartesian system and $\dot{x}_{1,t_2}$ and $\dot{x}_{2,t_2}$, the components of the target's velocity, all at absolute time t_2. The symbol $\mathbf{A}(t_2,t_1)$ denotes a known 4×4 matrix, defined by

$$\mathbf{A}(t_2,t_1) = \begin{bmatrix} \boldsymbol{I}_2 & (t_2-t_1)\boldsymbol{I}_2 \\ \mathbf{0}_2 & \boldsymbol{I}_2 \end{bmatrix},$$

where $t_2 - t_1 > 0$ is an absolute time offset, and $\boldsymbol{I}_2$ and $\mathbf{0}_2$ are the identity and zero 2×2 matrices, respectively. The state noise is represented by the 4×1 vector $\boldsymbol{u}_{t_2,t_1}$ whose distribution is known and assumed Gaussian with covariance matrix

$$\mathbf{Q}(t_2,t_1) = \sigma_u^2 \begin{bmatrix} \frac{(t_2-t_1)^3}{3} & \frac{(t_2-t_1)^2}{2} & 0 & 0 \\ \frac{(t_2-t_1)^2}{2} & (t_2-t_1) & 0 & 0 \\ 0 & 0 & \frac{(t_2-t_1)^3}{3} & \frac{(t_2-t_1)^2}{2} \\ 0 & 0 & \frac{(t_2-t_1)^2}{2} & (t_2-t_1) \end{bmatrix},$$

where σ_u^2 is the equivalent continuous time process noise intensity.

The target dynamics related to the agents' base time instants are

$$\begin{aligned} \boldsymbol{x}_{t,1} &= \mathbf{A}(t+\tau_1, t-1+\tau_N)\boldsymbol{x}_{t-1,N} + \boldsymbol{u}_{t+\tau_1, t-1+\tau_N} \\ \boldsymbol{x}_{t,i} &= \mathbf{A}(t+\tau_i, t+\tau_{i-1})\boldsymbol{x}_{t,i-1} + \boldsymbol{u}_{t+\tau_i, t+\tau_{i-1}}, \\ & \quad i = 2, \cdots, N, \end{aligned} \tag{2}$$

where N is the number of agents that track the target.

Within each time interval, the agents receive signals from the target as well as from other agents. The signal received by the i-th agent within the time interval from $t-1$ to t can be modeled by

$$\boldsymbol{y}_{t,i} = g_i(\boldsymbol{x}_{t,i}) + \boldsymbol{v}_{t,i}, \tag{3}$$

where $g_i(\cdot)$ is the measurement function given by

$$g_i(\boldsymbol{x}_{t,i}) = \frac{\Psi d_0^{\alpha}}{\|\boldsymbol{r}_{t,i} - \boldsymbol{l}_{t,i}\|^{\alpha}},$$

with $\boldsymbol{l}_{t,i} = [x_{1,t,i}\ x_{2,t,i}]^\top$ being the location of the target at time instant $(t-1)+\tau_i$; $\boldsymbol{r}_{t,i}$ denoting the location of the i-th agent at time instant $(t-1)+\tau_i$; Ψ representing the emitted signal power by the target measured at distance d_0; and α being a path-loss coefficient that depends on the transmission medium and assumed known. The observation noise $\boldsymbol{v}_{t,i}$ has known distribution and it does not have to be Gaussian. For simulation purposes, we assume that the noise is Gaussian with variance σ_v^2.

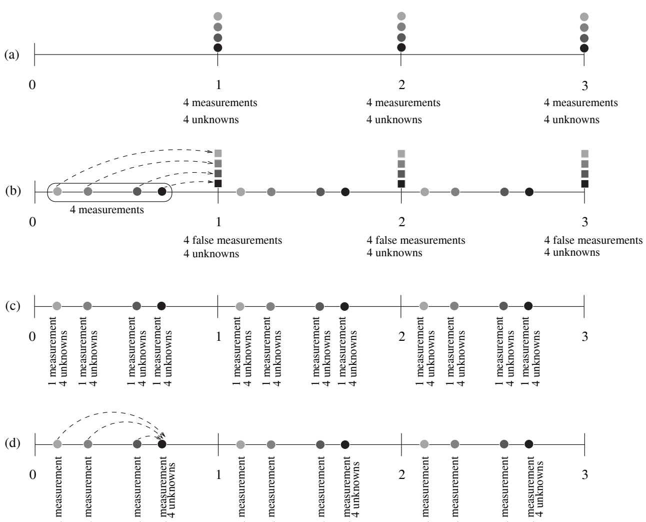

Fig. 1. (a) "True" synchronous method; (b) "false" synchronous method; (c) asynchronous sequential method; and (d) asynchronous batch method.

3. TRACKING ALGORITHMS

Each mobile agent performs tracking independently by employing particle filtering and by using its own measurements and the ones received from the other agents.[1] Therefore, unlike other distributed processing schemes where the agents exchange their estimates and the uncertainties about them, here the agents directly share their measurements. For comparison purposes we consider four different scenarios, which are illustrated in Fig. 1 for four mobile agents.

[1] Note that all the algorithms are applicable to scenarios with static agents.

(a) This scenario corresponds to a synchronous setup where all the agents receive the measurements at the same time. Therefore, at the tracking times of interest $(1, 2, \cdots)$ the agents have four measurements obtained at these times for tracking four unknowns (the position and velocity of the target). This scenario constitutes a benchmark for the performance of the remaining algorithms. The details of the method that tracks the target can be found in [6]. We refer to this method as *"true" synchronous method.*

(b) For this case, the agents receive the measurements at different time instants during a time interval. However, when processing the measurements for tracking, the agents assume that the observations were obtained synchronously, i.e., at the tracking times of interest $(1, 2, \cdots)$, there are four measurements obtained at different time instants but assumed to be obtained at the correct time for tracking four unknowns (the position and velocity of the target). We refer to this method as *"false" synchronous method.*

(c) Within this setup, the agents receive the measurements asynchronously during a time interval. As soon as they receive their measurement, they broadcast it and immediately update the position of the target. The agent also updates the filter upon receiving a measurement from another agent. In Fig. 1, within a time interval, there is a sequence of four measurements and for each of them the agents update the four unknowns. We refer to this method as *asynchronous sequential method.*

(d) In this situation, the agents also receive the measurements asynchronously during a time interval. However, they wait until they receive their own measurement to update the target parameters. In addition to their actual measurement, they also use the most recent observations from the remaining agents. Therefore, they account for the asynchronous nature of the measurements and use this knowledge to update their individual filters. We refer to this method as *asynchronous batch method.*

3.1. "False" synchronous method

This tracking setup ignores time differences between each of the agent's measurements, i.e., all measurements are wrongly assumed to be obtained at the same time instant. Although each of the agents form an estimate of the target location for their own base time $(t-1)+\tau_i$, it is assumed (incorrectly) that $(t-1)+\tau_i = k, k = 1, 2, \cdots$. In running the particle filter [8], each agent forms a random measure $\chi_k = \{\boldsymbol{x}_k^{(m)}, w_k^{(m)}\}_{m=1}^M$, where $\boldsymbol{x}_k^{(m)}$ are the particles and $w_k^{(m)}$ denote the weights associated to the particles. The particles for the i-th agent are propagated as

$$\boldsymbol{x}_k^{(m)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(m)}), \quad m = 1, 2, \cdots, M, \tag{4}$$

and the particle weights are computed according to

$$w_k^{(m)} \propto w_{k-1}^{(m)} \prod_{n=1}^{N} p\left(y_{t,n}|\boldsymbol{x}_k^{(m)}\right). \tag{5}$$

Note that M represents the number of particles that each agent uses for running its filter and for simplicity in notation it is assumed to be the same for each agent. Also note that if we assume that all agents initialize the filters in the same way and use the same seed for simulation, they all maintain the same random measure.

3.2. Asynchronous sequential method

In this case, each agent processes and updates its estimate of the target's location at its own base time as well as at all instances in which other agents' measurements are received. This solution therefore consists of filter updates using a single measurement at a time. The i-th agent maintains a random measure of the form $\chi_{t,i} = \{\boldsymbol{x}_{t,i}^{(m)}, w_{t,i}^{(m)}\}_{m=1}^{M}$ where $t = 1, 2, \cdots$ and $i = 1, \cdots, N$. The particles in each filter are propagated according to

$$\begin{aligned} \boldsymbol{x}_{t,1}^{(m)} &\sim p(\boldsymbol{x}_{t,1}^{(m)}|\boldsymbol{x}_{t-1,N}^{(m)}) \\ \boldsymbol{x}_{t,i}^{(m)} &\sim p(\boldsymbol{x}_{t,i}^{(m)}|\boldsymbol{x}_{t,i-1}^{(m)}), \qquad i = 2, \cdots, N, \end{aligned} \tag{6}$$

and the particle weights of each agent at each base time instant $(t-1)+\tau_i$ follow the expression

$$\begin{aligned} w_{t,1}^{(m)} &\sim w_{t-1,N}^{(m)} p\left(y_{t,1}|\boldsymbol{x}_{t,1}^{(m)}\right) \\ w_{t,i}^{(m)} &\sim w_{t,i-1}^{(m)} p\left(y_{t,i}|\boldsymbol{x}_{t,i}^{(m)}\right), \qquad i = 2, \cdots, N. \end{aligned} \tag{7}$$

3.3. Asynchronous batch method

For this solution, each agent uses it own measurement as well as the most recently available measurements from other agents to form an estimate of the target's location only at its own base time. Since the i-th agent no longer directly tracks the target location at other base time instants $(t-1)+\tau_n$, $n \neq i$, it forms an estimate of the target location at these other instants based on the particle at its own base time instant $(t-1)+\tau_i$

$$\hat{\boldsymbol{x}}_{t,n}^{(m)} = E\left[\boldsymbol{x}_{t,n}|\boldsymbol{x}_{t,i}^{(m)}\right]. \tag{8}$$

The particles for the i-th agent are propagated as

$$\boldsymbol{x}_{t,i}^{(m)} \sim p(\boldsymbol{x}_{t,i}|\boldsymbol{x}_{t-1,i}^{(m)}), \quad m = 1, 2, \cdots, M, \tag{9}$$

and the corresponding particle weights are updated at its own base time only according to

$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} p\left(y_{t,i}|\boldsymbol{x}_{t,i}^{(m)}\right) \prod_{n=1, n\neq i}^{N} p\left(y_{t,n}|\hat{\boldsymbol{x}}_{t,n}^{(m)}\right). \tag{10}$$

This solution is labeled as *batch-1* in the simulation section.

We also propose an alternative to the previous solution which introduces the concept of an "asynchronous particle filter" and is based on [6]. The conventional representation of the filtering density of the target state (estimated for sensor i at its base time $(t-1)+\tau_i$ can be expressed as $p\left(\boldsymbol{x}_{t,i}|y_{1:t,1:N}\right)$ where in this form each of the $y_{t,i}$ represents a single measurement at time $(t-1)+\tau_i$ by sensor i. This form is modified to read $p\left(\boldsymbol{x}_{t,i}|\boldsymbol{y}_{1:t}\right)$. Single scalar measurements have now been replaced with measurement vectors. A given vector $\boldsymbol{y}_t$ represents the collection of measurements that were made in the time interval from $t-1$ to t and is expressed as $\boldsymbol{y}_t = [y_{t,1}, y_{t,2}, \cdots, y_{t,N}]$.

Whereas in the sequential processing solution the estimate of the state (and corresponding particles weights) is updated at the time of each new measurement, this solution attempts to estimate the state only at a given sensor's own base time instants. With this in mind, the likelihood function which is used in updating the particle weights, can no longer be expressed as $p\left(\boldsymbol{y}_t|\boldsymbol{x}_{t,i}\right)$ since the measurements within the vector $\boldsymbol{y}_t$ do not depend only on the state at time $(t-1)+\tau_i$ but also on "intermediate" states within the time interval from $t-1$ to t. As such, the likelihood function is modified to $p\left(\boldsymbol{y}_t|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right)$. The new expression for updating the weights of each particle of the i-th agent is

$$w_{t,i}^{(m)} \propto w_{t-1,i}^{(m)} p\left(\boldsymbol{y}_t|\boldsymbol{x}_{t-1,i}^{(m)}, \boldsymbol{x}_{t,i}^{(m)}\right). \tag{11}$$

It has been shown that the modified or "asynchronous" likelihood can be approximated as

$$p\left(\boldsymbol{y}_t|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right) \approx \prod_{n=1}^{N} p\left(y_{t,n}|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right), \tag{12}$$

and each term can be decomposed according to

$$p\left(y_{t,n}|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right) = \int p\left(y_{t,n}|\boldsymbol{x}_{t,n}\right) p\left(\boldsymbol{x}_{t,n}|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right) d\boldsymbol{x}_{t,n}. \tag{13}$$

In either case, it is difficult to evaluate the "asynchronous" likelihood without using further approximations. With the given model for the target state dynamics, the term $p\left(\boldsymbol{x}_{t,n}|\boldsymbol{x}_{t-1,i}, \boldsymbol{x}_{t,i}\right)$ is a known Gaussian and will be sharply peaked as long as the target state process noise intensity σ_u^2 is reasonably small. As such, we can form a fairly accurate approximation of this integral using Monte Carlo integration with relatively few sample points. In summary, the algorithm proceeds as follows:

Step 1: For the i-th agent, initialize all weights at time $(t-1)+\tau_i$

$$w_{t,i}^{(m)} \propto p\left(y_{t,i}|\boldsymbol{x}_{t,i}^{(m)}\right).$$

Step 2: Propagate the particles $\boldsymbol{x}_{t,i}^{(m)} \sim p(\boldsymbol{x}_{t,i}|\boldsymbol{x}_{t-1,i}^{(m)})$.

Step 2a: Repeat Step 2 for all other agents $j \neq i$, generate K samples

$$\hat{\boldsymbol{x}}_{t,j}^{(k,m)} \sim p(\boldsymbol{x}_{t,j}|\boldsymbol{x}_{t,i}^{(m)}), \quad k = 1, 2, \cdots, K.$$

Step 2b: Evaluate the integrand in (13) at each sample

$$\hat{w}_{t,j,i}^{(k,m)} = p\left(y_{t,j}|\hat{\boldsymbol{x}}_{t,j}^{(k,m)}\right) p\left(\hat{\boldsymbol{x}}_{t,j}^{(k,m)}|\boldsymbol{x}_{t-1,i}^{(m)}, \boldsymbol{x}_{t,i}^{(m)}\right).$$

Step 2c: Update the main particle weights as $w_{t,i}^{(m)} \propto w_{t,i}^{(m)} \sum_{n=1}^{K} \hat{w}_{t,j,i}^{(k,m)}$.

Step 3: Resample if necessary.

This solution is labeled as *batch-2* in the simulations section.

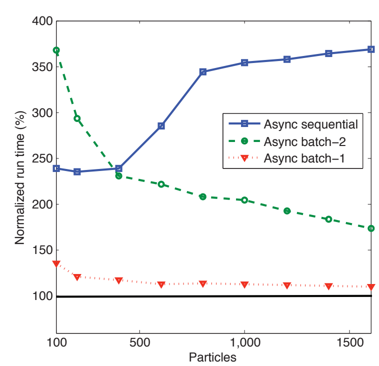

Fig. 2. Normalized run-time as a function of the number of particles.

4. SIMULATION RESULTS

Nine mobile agents are initially distributed across a 3 x 3 grid. The target approaches the sensor grid from a number of possible locations with a fixed initial velocity and subject to random perturbations. Once the target is within detection range, four of the agents are assigned to track the target. Each of the agents obtains measurements of the target at different time offsets from one another within a given sample period. We consider a scenario without interference sources and sudden maneuvers. We also assume that the agents can infer the location of the other agents exactly. We set the process noise intensity σ_u^2 to 0.005 and measurement noise variance σ_v^2 to $.005$ for all experiments. Other values of σ_v^2 were tested and it was found that as σ_v^2 increases, the asynchronism of the measurements becomes less relevant for the performance since the error due to the noise becomes larger than the error due to incorrectly assuming that the measurements occurred at the same instant. The deployment of the agents is on a ball centered at the predicted location of the target and a radius of 3m.

Figure 2 shows the average simulation run-time (per time interval) for the three asynchronous solutions for different total number of particles used to run the algorithms. The normalized run time was defined as the ratio between the run-time per time interval of the considered algorithm over the run-time per time interval obtained by the "true" synchronous method as a reference.[2] For the batch-2 method 10 particles were used for the integration step. It is clear from the figure that the batch-1 method is the best in terms of computational load, and does not require much additional processing time compared to the "true" synchronous method. Also the sequential method is faster than the batch-2 algorithm only for a low number of particles. This is expected as the processing time for the sequential method is more dependent on the particle count, while the batch-2 method requires approximately the same amount of "overhead" regardless of the number of particles.

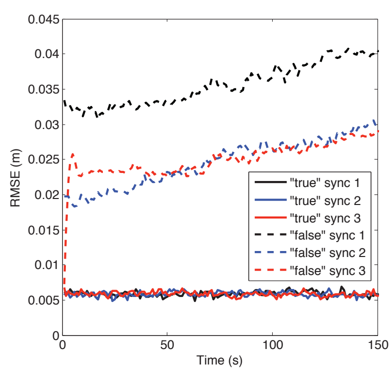

Fig. 3. RMSE performance for different sets of measurement times.

For the next experiment we considered three different scenarios corresponding to three different sets of measurement times, i.e., scenario 1 corresponds to a set of time delays given by $\boldsymbol{\tau} = \{\tau_i\}_{i=1}^4 = \{0, 0.25, 0.5, 0.75\}$; scenario 2 by $\boldsymbol{\tau} = \{0, 0.1, 0.5, 0.51\}$; and scenario 3 by $\boldsymbol{\tau} = \{0, 0.05, 0.15, 0.75\}$. Figure 3 illustrates the comparison in performance for each of the sets for the "true" synchronous method and the "false" synchronous method. It can be seen that scenario 1, corresponding to evenly spaced measurements throughout the interval, yields the worst performance for the "false" synchronous method (note that the "true" synchronous algorithm does not get affected by the different scenarios because the measurements are always at integral times). Although the performance is also degraded with the remaining two sets of measurement times, there is no significant difference between them.

Finally, we also simulated the various asynchronous solutions for the previous scenario 1 of measurement times. The results in Fig. 4 show how the proposed solutions correctly account for the asynchronous nature of the measurements and perform similar to the "true" synchronous algorithm.

[2]Note that a value of 100% denotes equal time of execution of the algorithm with respect to the "true" synchronous method.

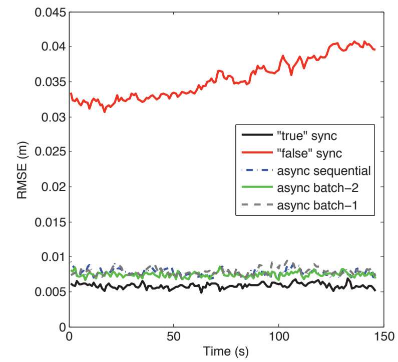

Fig. 4. RMSE performance for all the algorithms for scenario 1 given by $\boldsymbol{\tau} = \{0, 0.25, 0.5, 0.75\}$.

5. CONCLUSIONS

In this paper we propose a number of non-centralized solutions to deal with asynchronous measurements for target tracking with mobile agents. The results obtained with the proposed methods which account for the asynchronism of the measurement reveal significant improvement compared to a method that assumes synchronism. In addition, we introduce an algorithm that accounts for the asynchronism and yet does not require a significant increase in computational load relative to the "true" synchronous tracking solution.

6. REFERENCES

[1] K. Dantu, M. Rahimi, H. Shah, S. Babeland, A. Dhariwal, and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," in *IPSN '05: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005.

[2] Y. Li and P. M. Djurić, "Particle filtering for target tracking with mobile sensors," in *the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[3] Y. Li and P. M. Djurić, "Target tracking with mobile sensors using cost-reference particle filtering," in *the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

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

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

[6] P. M. Djurić, J. Beaudeau, and M. F. Bugallo, "Non-centralized target tracking with mobile agents," in *the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[7] A. F. Garcia-Fernández and J. Grajal, "Asynchronous particle filter for tracking using non-synchronous sensor networks," *Signal Processing*, vol. 91, pp. 2304–2313, 2011.

[8] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.