A MONTE CARLO METHOD FOR JOINT NODE LOCATION AND MANEUVERING TARGET TRACKING IN A SENSOR NETWORK

Joaquín Míguez and Antonio Artés-Rodríguez.

Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911 Leganés, Madrid (Spain)
E-mail: `jmiguez@ieee.org, antonio@ieee.org`

ABSTRACT

We address the problem of tracking a maneuvering target that moves along a region monitored by a sensor network, whose nodes (including both the sensors and any data fusion centers, DFCs) are located at unknown positions. Thus, the node locations and the target track *must be estimated jointly without the aid of beacons*. We assume that the network consists of a collection of sensors and at least four DFCs. Each DFC collects and integrates the sensor measurements and can exchange data with the other DFCs. Within this setup, we propose a three-stage Monte Carlo method to (i) acquire rough initial estimates of the network node locations, (ii) track the target and refine the node position estimates individually at each DFC and (iii) fuse the results obtained by all the DFCs. The validity of the method is illustrated by computer simulations of a network of power-aware sensors and exactly four DFCs.

1. INTRODUCTION

Sensor networks will soon become ubiquitous because of their suitability for a broad range of emerging applications, such as environmental monitoring, surveillance and security, vehicle navigation, tracking, logistics, etc... For virtually any of these applications, the accurate localization of the sensors is a key task. Indeed, automatic node positioning has been recognized as an enabling technology, since the data measured by a sensor is hardly useful unless it is precisely known *where* it has been collected [1]. Most sensor localization algorithms rely on the availability of beacons, i.e., network nodes with known position that can be taken as reference [1]. Although beacon-free network designs are feasible [2], they usually involve complicated energy-consuming local communications among nodes that should ideally be simple and devoted to sensing and transmitting data.

In this paper, we address the problem of using a network of distance-aware sensors with unknown locations to track a maneuvering target. We assume that there are no beacons available, which implies that the positions of the control nodes in the network, that we will hereafter term data fusion centers (DFCs), are also unknown. We consider networks in which the sensors can only collect and broadcast measurements, while there are at least four DFCs with (point-to-point) communication and sensing capabilities. Within this setup, we propose a three-stage Monte Carlo method to (i) acquire rough initial estimates of the network node locations, (ii) track the target and refine the node position estimates individually at each DFC and (iii) coherently combine the results obtained by the DFCs.

The remaining of the paper is organized as follows. In Section 2, we provide a mathematical model of the class of systems under consideration. The proposed algorithm is described in Section 3. In Section 4 we present illustrative computer simulation results for a network of power-aware sensors and exactly four DFCs. Finally, Section 5 is devoted to the conclusions.

This work was supported by Xunta de Galicia (project PGIDIT04TIC105008PR), Ministerio de Educación y Ciencia of Spain (project 'DOIRAS', id. TIC2003-02602) and the UE (Network of Excellence 'CRUISE', id. IST-4-027738).

2. SYSTEM MODEL

We assume that the target moves along a 2-dimensional region according to the linear model [3]

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{q}u_t, \quad t = 1, 2, 3, \dots \qquad (1)$$

where $\mathbf{x}_t = [r_t, v_t]^\top \in \mathbb{C}^2$ is the target state, which includes its position and its velocity at time t, $r_t \in \mathbb{C}$ and $v_t \in \mathbb{C}$, respectively; $\mathbf{A} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix}$ is a transition matrix that depends on the observation period, T_s; $\mathbf{q} = [T_s^2/2, T_s]^\top$ and $u_t \sim CN(u_t; 0, \sigma_u^2)$ is a complex Gaussian noise term with zero mean and variance σ_u^2. The initial target state, $\mathbf{x}_0$, has a known prior probability density function (pdf), $p(\mathbf{x}_0)$.

The N_s sensors in the network are located at fixed unknown positions $s_{1:N_s} := \{s_1, s_2, \dots, s_{N_s}\}$ with uniform prior pdf on the region of interest, $\mathcal{R}$, i.e., $s_n \sim U(\mathcal{R})\ \forall n$. The n-th sensor measurement at time t is denoted as $y_{n,t} = f_s(d_{n,t}, \epsilon)$, where $d_{n,t} = |r_t - s_n|$ is the distance between the target and the sensor and ϵ is a random perturbation with known pdf. An $N_t \times 1$ vector of measurements, $\mathbf{y}_t := [y_{\kappa(1),t}, \dots, y_{\kappa(N_t),t}]^\top$, where $\kappa(i) \in \{1, \dots, N_s\}$, is broadcast to the DFCs at time t. Note that not every sensor transmits at every time. Indeed, it is often convenient (in order to reduce energy consumption) that only a subset of sensors become active and transmit their measurement, hence $N_t \leq N_s$.

The n-th DFC, $n = 1, ..., N_c$, is located at the unknown position $r_n^o \in \mathbb{C}$, with known prior pdf $p(r_n^o)$, and collects N_t sensor measurements at time t. We also assume that the DFC has the capability to extract some distance related magnitude from the communication signals transmitted by the sensors. For simplicity, let us assume the same type of measurement carried out at the sensors, hence the n-th DFC also has at its disposal the $N_t \times 1$ data vector $\mathbf{z}_{n,t} = [z_{n,\kappa(1),t}, \dots, z_{n,\kappa(N_t),t}]^\top$, where $z_{n,i,t} = f_s(d_{n,i,t}^c, \epsilon)$, $d_{n,i,t}^c = |s_{i,t} - r_n^o|$ is the distance between the i-th sensor and the n-th DFC and ϵ is a random perturbation.

Our goal is to jointly estimate the target track $\mathbf{x}_{0:t} := \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$, the sensor locations, $s_{1:N_s}$, and the DFC positions, $r^o_{1:N_c}$, from the sequence of data vectors $\mathbf{y}_{1:t} := \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ and $\mathbf{z}_{1:N_c,1:t} := \{\mathbf{z}_{1,1}, \ldots, \mathbf{z}_{1,t}, \mathbf{z}_{2,1}, \ldots, \mathbf{z}_{N_c,t}\}$ without the aid of beacons. Note that, because of the use of distance-aware measurements and the lack of any absolute reference position, the estimates will be subject to an inherent rotation ambiguity.

3. ALGORITHM

The proposed method consists of three stages, that we outline below.

Stage 1. Acquisition: Initial rough estimates of $s_{1:N_s}$ and $r^o_{1:N_c}$ are computed by (approximately) solving the optimization problems

$$\hat{r}^o_{1:N_c,0} = \arg\max_{\tilde{r}^o_{1:N_c}} \left\{ \prod_{i \neq l} p(\pi^c_{il} | \tilde{r}^o_i, \tilde{r}^o_l) p(\tilde{r}^o_{1:N_c}) \right\} \quad (2)$$

$$\hat{s}_{n,0} = \arg\max_{\tilde{s}_n} \left\{ \prod_{l=1}^{N_c} p(\pi^s_{nl} | \hat{r}^o_{n,0}, \tilde{s}_n) \right\}, \quad (3)$$

$n = 1, ..., N_s$, where π^c_{il} is a measurement obtained at the l-th DFC from a signal transmitted by the i-th DFC (hence, $i \neq l$) and π^s_{nl} is a measurement obtained at the l-th DFC from a signal transmitted by the n-th sensor. Apparently, this step requires that each DFC and each sensor transmit at least one signal burst that can be collected by the other DFC nodes in the network. Also note that both $\hat{r}^o_{1:N_c,0}$ and $\hat{s}_{n,0}$ are maximum a posteriori (MAP) type of estimators (although the prior $p(s_n)$ is uniform, and hence omitted, in (3)).

Stage 2. Tracking: We propose an auxiliary particle filter (APF) algorithm for state estimation in dynamic systems with unknown fixed parameters derived according to [4]. An APF is independently run at each DFC so that, at any prescribed time t, we obtain N_c approximately-MAP estimates, specifically $\left\{ \mathbf{x}^{MAP}_{0:t}(i), s^{MAP}_{1:N_s,t}(i), r^{o,MAP}_{i,t}(i) \right\}_{i=1}^{N_c}$ (note that each DFC estimates its own location). The efficiency of the APF is improved by drawing the initial set of particles (samples in the space of $(\mathbf{x}_0, s_{1:N_s}, r^o_{1:N_c})$) in a neighborhood of the initial estimates $\hat{r}^o_{1:N_c,0}$ and $\hat{s}_{n,0}$.

Stage 3. Fusion of estimates: The approximately-MAP estimates produced by the DFCs are fused taking into account that they may have different rotations (because of the insensitivity of distance-aware measurements to the angle between the transmit and receive nodes). Thus, we keep the estimate from the first FDC fixed and rotate the others by angles

$$\hat{\theta}_{i,t} = \arg\min_{\theta \in [0, 2\pi)} \left\{ \sum_{k=0}^{t} \left\| e^{j\theta} \mathbf{x}^{MAP}_k(i) - \mathbf{x}^{MAP}_k(1) \right\|_2^2 \right\}, \quad (4)$$

$i = 2, 3, 4$, where $j = \sqrt{-1}$, i.e., we rotate the track estimates of DFCs 2, 3 and 4 to minimize the mismatch with the track estimate of DFC 1.

3.1. Acquisition

We assume that the π^c_{il} and π^s_{nl} measurements used in (2) and (3), respetively, are also functions of distance, namely $\pi^c_{il} = f_c(|r^o_i - r^o_l|, \epsilon)$ and $\pi^s_{nl} = f_c(|s_n - r^o_l|, \epsilon)$ for some function f_c and random perturbations ϵ. The difficulty for solving (2) is the relatively large dimensionality of the problem, since $N_c \geq 4$ complex parameters must be jointly optimized. On the contrary, once $\hat{r}_{1:N_c,0}$ has been computed, the positions of the sensors can be selected independently.

For "static" problems like (2) and (3) (which have no simple analytic solution), we propose to apply the accelerated random search (ARS) algorithm [5] described in Table 1 for a general minimization problem (its application to maximization is straightforward).

Problem: $\hat{\alpha} = \arg\min_{\alpha \in \mathcal{A}} g(\alpha)$ for some function g. Denote: $r_{min} > 0$, the "minimum radius"; $r_{max} > r_{min}$, the "maximum radius"; $r_{max} \geq r_n \geq r_{min}$ the radius at the n-th iteration; $c > 1$ the "contraction" factor; α_n the solution obtained after the n-th iteration; and $B_n := \{\tilde{\alpha} \in \mathcal{A} : \|\tilde{\alpha} - \alpha_n\|_2 < r_n\}$, where $\| \cdot \|_2$ indicates 2-norm. **Algorithm:** given r_n and α_n, (1) Draw $\tilde{\alpha} \sim U(B_n)$. (2) If $g(\tilde{\alpha}) < g(\alpha_n)$ then $\alpha_{n+1} = \tilde{\alpha}$ and $r_{n+1} = r_{max}$, else $\alpha_{n+1} = \alpha_n$ and $r_{n+1} = r_n/c$. (3) If $r_{n+1} < r_{min}$, then $r_{n+1} = r_{max}$. (4) Go back to (1)

Table 1. Iterative ARS algorithm for a minimization problem. Parameter α is possibly multidimensional (typically, $\alpha \in \mathbb{C}^n$).

3.2. Tracking

The aim is to track the sequence of states $\mathbf{x}_{0:t}$, and improve the estimation of $s_{1:N_t}$ and $r^o_{1:N_c}$, given the measurements $\mathbf{y}_{1:t}$ and $\mathbf{z}_{1:N_c,1:t}$. We propose to use an APF algorithm based on [4] at each DFC. The APF is a recursive algorithm that generates a sequence of discrete probability measures, denoted $\Omega_t = \{(\mathbf{x}_t, s_{1:N_s,t}, r^o_{n,t})^{(i)}, w^{(i)}_t\}_{i=1}^M$, that approximate the posterior pdf's of the unknowns, i.e., for $n \in \{1, \ldots, N_c\}$,

$$p(\mathbf{x}_t, s_{1:N_s}, r^o_n | \mathbf{y}_{1:t}, \mathbf{z}_{n,1:t}) \approx p_M(\mathbf{x}_t, s_{1:N_s}, r^o_n | \mathbf{y}_{1:t}, \mathbf{z}_{n,1:t}) = \sum_{i=1}^M \delta_i(\mathbf{x}_t, s_{1:N_s}, r^o_n) w^{(i)}_t, \quad (5)$$

where $\delta_i(\mathbf{x}_t, s_{1:N_s}, r^o_n) = 1$ if $(\mathbf{x}_t, s_{1:N_s}, r^o_n) = (\mathbf{x}_t, s_{1:N_s,t}, r^o_{n,t})^{(i)}$ and 0 otherwise. The samples, $\mathbf{x}^{(i)}_t$, $s^{(i)}_{1:N_s,t}$ and $r^o_{n,t}{}^{(i)}$ (for $i = 1, ..., M$) are called *particles* and the probabilities $w^{(i)}_t$ (note that $\sum_{i=1}^M w^{(i)}_t = 1$) are called *weights*. When the time t measurements, $\mathbf{y}_t$ and $\mathbf{z}_{n,t}$, become available, Ω_t is recursively computed from Ω_{t-1} as indicated in Table 2.

The proposed APF algorithm is based on the relationship

$$p(\mathbf{x}_t, s_{1:N_s}, r^o_n | \mathbf{y}_{1:t}, \mathbf{z}_{n,1:t}) \propto p(\mathbf{y}_t, \mathbf{z}_{n,t} | \mathbf{x}_t, s_{1:N_s}, r^o_n) \times \times p(\mathbf{x}_t | s_{1:N_s}, \mathbf{y}_{1:t-1}) p(s_{1:N_s}, r^o_n | \mathbf{y}_{1:t-1}, \mathbf{z}_{n,1:t-1}) \quad (6)$$

and the approximations

$$p_M(\mathbf{x}_t | s_{1:N_s}, \mathbf{y}_{1:t-1}) = \sum_{i=1}^M p(\mathbf{x}_t | \mathbf{x}^{(i)}_{t-1}) \delta_i(s_{1:N_s}) w^{(i)}_{t-1} \quad (7)$$

$$p_M(s_{1:N_s}, r^o_n | \mathbf{y}_{1:t-1}, \mathbf{z}_{1:t-1}) = \sum_{i=1}^M w^{(i)}_{t-1} K_i(s_{1:N_s}, r^o_n), \quad (8)$$

where $K_i(\cdot)$ is a symmetric kernel. For the latter, we have chosen

$$K_i(s_{1:N_s}, r^o_n) = CN(s_{1:N_s}, r^o_n | \boldsymbol{\mu}^{(i)}_{t-1}, h^2 \boldsymbol{\Sigma}_{t-1}) \quad (9)$$

$t=0$: Draw $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0)$, $s_{1:N_s,0}^{(i)} \sim N(s_{1:N_s}\|\hat{s}_{1:N_s,0},\sigma_{s,0}^2)$ and ${r_{n,0}^o}^{(i)} \sim N(r_n^o\|\hat{r}_{n,0}^o,\sigma_{r,0}^2)$.
t: Given $\Omega_{t-1} = \left\{ \left(\mathbf{x}_{t-1}, s_{1:N_s,t-1}, r_{n,t-1}^o\right)^{(i)}, w_{t-1}^{(i)} \right\}_{i=1}^M$:
(1) Compute $\tilde{\mathbf{x}}_t^{(i)} = \mathbf{A}\mathbf{x}_{t-1}^{(i)}$, $i=1,...,M$.
(2) Draw indices $\ell^{(i)} \sim q_t(\ell)$, $i=1,...,M$, where $q_t(\ell) \propto {w_{t-1}^{(\ell)}}^2 p(\mathbf{y}_t,\mathbf{z}_{n,t}\|\tilde{\mathbf{x}}_t^{(\ell)}, s_{1:Ns,t-1}^{(\ell)}, {r_{n,t-1}^o}^{(\ell)})$.
(3) Draw $\left(s_{1:N_s,t}, r_{n,t}^o\right)^{(i)} \sim q_t(s_{1:N_s}, r_n^o\|\ell^{(i)})$, where $q_t(s_{1:N_s}, r_n^o\|\ell^{(i)}) = N(s_{1:N_s}, r_n^o\|\boldsymbol{\mu}_{t-1}^{(\ell^{(i)})}, h^2\boldsymbol{\Sigma}_{t-1})$, for $i=1,...,M$.
(4) Draw target states $\mathbf{x}_t^{(i)} \sim p(\mathbf{x}_t\|\mathbf{x}_{t-1}^{(\ell^{(i)})})$, $i=1,...M$, and build the trajectory $\mathbf{x}_{0:t}^{(i)} = \{\mathbf{x}_{0:t-1}^{(\ell^{(i)})}, \mathbf{x}_t^{(i)}\}$.
(5) Update the weights, for $i=1,...,M$, $w_t^{(i)} \propto \frac{p\left(\mathbf{y}_{1:t},\mathbf{z}_{n,1:t}\|\mathbf{x}_t^{(i)}, s_{1:N_s,t}^{(i)}, {r_{n,t}^o}^{(i)}\right)}{p\left(\mathbf{y}_{1:t},\mathbf{z}_{n,1:t}\|\left(\tilde{\mathbf{x}}_t, s_{1:N_s,t-1}, r_{n,t-1}^o\right)^{(\ell^{(i)})}\right)}$.
(6) MAP estimation, $i_o = \arg\min_{i\in\{1,...,M\}}\{w_t^{(i)}\}$, $\left(\mathbf{x}_t^{MAP}(n), s_{1:N_s,t}^{MAP}(n), r_{n,t}^{o,MAP}(n)\right) = \left(\mathbf{x}_{0:t}, s_{1:N_s,t}, r_{n,t}^o\right)^{(i_o)}$.

Table 2. Liu and West's APF algorithm for joint estimation of the target trajectory, $\mathbf{x}_{0:t}$, and the fixed node locations, $s_{1:N_s}$ and r_n^o from the measurements available at the n-th DFC.

where

$$\boldsymbol{\mu}_{t-1}^{(i)} = [\bar{s}_{1,t-1}^{(i)}, \ldots, \bar{s}_{N_s,t-1}^{(i)}, {\bar{r}_{n,t-1}^o}^{(i)}]^\top \quad (10)$$

$$\boldsymbol{\Sigma}_{t-1} = \text{diag}\{\sigma_{1,t-1}^2, \ldots, \sigma_{N_s,t-1}^2, \sigma_{r,t-1}^2\} \quad (11)$$

and $h \in (0,1)$ is a bandwidth factor. The kernel modes are calculated as ${\bar{r}_{n,t-1}^o}^{(i)} = a{r_{n,t-1}^o}^{(i)} + (1-a)\bar{r}_{n,t-1}^o$ and $\bar{s}_{k,t-1}^{(i)} = as_{k,t-1}^{(i)} + (1-a)\bar{s}_{k,t-1}$, for $a = \sqrt{1-h^2}$ and $\bar{r}_{n,t-1}^o = \sum_{k=1}^M w_{t-1}^{(i)} {r_{n,t-1}^o}^{(i)}$, $\bar{s}_{k,t-1} = \sum_{k=1}^M w_{t-1}^{(i)} s_{k,t-1}^{(i)}$. The variances, in turn, are found as $\sigma_{r,t-1}^2 = \sum_{l=1}^M w_{t-1}^{(l)} \left|{r_{n,t-1}^o}^{(l)} - \bar{r}_{n,t-1}^o\right|^2$ and $\sigma_{k,t-1}^2 = \sum_{l=1}^M w_{t-1}^{(l)} \left|s_{k,t-1}^{(l)} - \bar{s}_{k,t-1}\right|^2$. This choice of $\boldsymbol{\mu}_{t-1}^{(i)}$ and $\boldsymbol{\Sigma}_{t-1}$ ensures that the mean and marginal variance of every fixed parameter given by the kernel approximation (8) is equal to the corresponding mean and marginal variance given by the weights [4].

One difficulty with the approximations (7) and (8) is that they involve mixtures of a typically large number (M) of pdf's. We avoid this limitation by incorporating a discrete auxiliary random variable $\ell \in \{1,...,M\}$ that indicates the terms in (7) and (8) to be selected. In particular, we define

$$p(\mathbf{x}_t, s_{1:N_s}, r_n^o, \ell|\mathbf{y}_{1:t}, \mathbf{z}_{n,1:t}) \propto p(\mathbf{y}_t, \mathbf{z}_{n,t}|\mathbf{x}_t, s_{1:N_s}, r_n^o) \times \times p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(\ell)}) {w_{t-1}^{(\ell)}}^2 K_\ell(s_{1:N_s}, r_n^o). \quad (12)$$

Using (12) we can easily draw particles and compute weights by applying the principle of importance sampling (IS) [6]. In particular, we define a suitable *importance function*, or proposal pdf,

$$q_t(\mathbf{x}_t, s_{1:N_s}, r_n^o, \ell) = q_t(\ell) q_t(s_{1:N_s}, r_n^o|\ell) p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(\ell)}) \quad (13)$$

(see Table 2 for the details) that we use for drawing new particles and then update the weights as

$$w_t^{(i)} \propto \frac{p\left(\left(\mathbf{x}_t, s_{1:N_s,t}, r_{n,t}^o, \ell\right)^{(i)} |\mathbf{y}_{1:t}, \mathbf{z}_{n,1:t}\right)}{q_t\left(\left(\mathbf{x}_t, s_{1:N_s,t}, r_{n,t}^o, \ell\right)^{(i)}\right)}, i = 1, \ldots, M. \quad (14)$$

The auxiliary variables are discarded before proceeding to time $t+1$.

We finally note that, given Ω_t, it is straightforward to produce estimates of the target trajectory and the node locations (in particular, it is enough to select the particle with the largest weight, as shown in Table 2). Thus, at any given time t, the n-th DFC can produce an approximate MAP estimate of $\mathbf{x}_{0:t}$, $s_{1:N_s}$ and r_n^o.

3.3. Fusion of estimates

After the second stage, we have obtained estimates $\left(\mathbf{x}_{0:t}^{MAP}(n), s_{1:N_s,t}^{MAP}(n), r_{n,t}^{o,MAP}(n)\right)$, $n = 1,...,N_c$, which are not necessarily coherent because of the insensitivity to rotations of the measurements $\mathbf{y}_{1:t}$ and $\mathbf{z}_{n,1:t}$. Therefore, we search for adequate phase rotations $\theta_{n,t}$, $n = 1,...,N_c$, that enable coherent combination, by solving the set of problems given by (4). The same as in the acquisition step, we may use the ARS algorithm to search the desired solutions. Once they are found, the N_c MAP estimates are fused to yield $\mathbf{x}_{0:t}^{MAP} := \sum_{n=1}^{N_c} e^{j\hat{\theta}_{n,t}} \mathbf{x}_{0:t}^{MAP}(n)$, $s_{1:N_s,t}^{MAP} := \sum_{n=1}^{N_c} e^{j\hat{\theta}_{n,t}} s_{1:N_s,t}^{MAP}(n)$ and $r_{n,t}^{o,MAP} := e^{j\hat{\theta}_{n,t}} r_{n,t}^{o,MAP}(n)$, where $\hat{\theta}_{1,t} = 0$.

4. COMPUTER SIMULATIONS

In order to provide illustrative numerical results, we have particularized the model of Section 2 to a network of power-aware sensors. Thus, the measurement functions f_s and f_c are equal,

$$f_s(d,\epsilon) = f_c(d,\epsilon) = 10\log_{10}\left(\frac{1}{d^2} + \eta\right) + \epsilon, \quad \text{(dB)} \quad (15)$$

where $\epsilon \sim N(\epsilon|0,1)$ and $\eta = 10^{-6}$ accounts for the power of the background noise (-60 dB). The n-th sensor transmits its measurement, $y_{n,t}$, only if it corresponds to a distance $d_{n,t} < 50$ m (i.e., $y_{n,t} > -33.97$ dB) and otherwise remains silent. The noise term is standard Gaussian, $\epsilon \sim N(\epsilon|0,1)$. As a consequence, all likelihood functions are Gaussian, namely

$$p(\mathbf{y}_t, \mathbf{z}_{n,t}|\tilde{\mathbf{x}}_t, \tilde{s}_{1:N_s}, \tilde{r}_n^o) = \prod_{i=1}^{N_t} N\left(y_{\kappa(i),t}|f_s(|\tilde{r}_t - \tilde{s}_{\kappa(i)}|, 0), 1\right) \times \times N\left(z_{n,\kappa(i),t}|f_s(|\tilde{r}_n^o - \tilde{s}_{\kappa(i)}|, 0), 1\right), \quad (16)$$

$$p(\pi_{il}^c|\tilde{r}_i^o, \tilde{r}_l^o) = N\left(\pi_{il}^c|f_c(|\tilde{r}_i^o - \tilde{r}_l^o|, 0), 1\right), \quad (17)$$

$$p(\pi_{nl}^s|\hat{r}_{l,0}^o, \tilde{s}_n) = N\left(\pi_{nl}^s|f_c(|\tilde{s}_n - \hat{r}_{l,0}^o|, 0), 1\right). \quad (18)$$

Since the prior $p(r_{2:4}^o)$ is also Gaussian, it is apparent that (2) and (3) become nonlinear least squares minimization problems.

There are $N_c = 4$ DFCs and $N_s = 20$ sensors in the network. We assume $r_1^o = 0$, while the others are a priori distributed as $r_{2:4}^o \sim CN\left(30 \times [-1, 1+j, 1-j]^\top, 25\mathbf{I}_3\right)$. This prior pdf is used to draw an initial guess of $r_{2:4}^o$ which is used as input to the ARS algorithm that solves (2), the other parameters being $c=2$, $r_{max} = 10$, $r_{min} = 10^{-4}$. The ARS algorithm for problem (3) receives as inputs a sensor position drawn from $U(\mathcal{R})$ (where $\mathcal{R}$

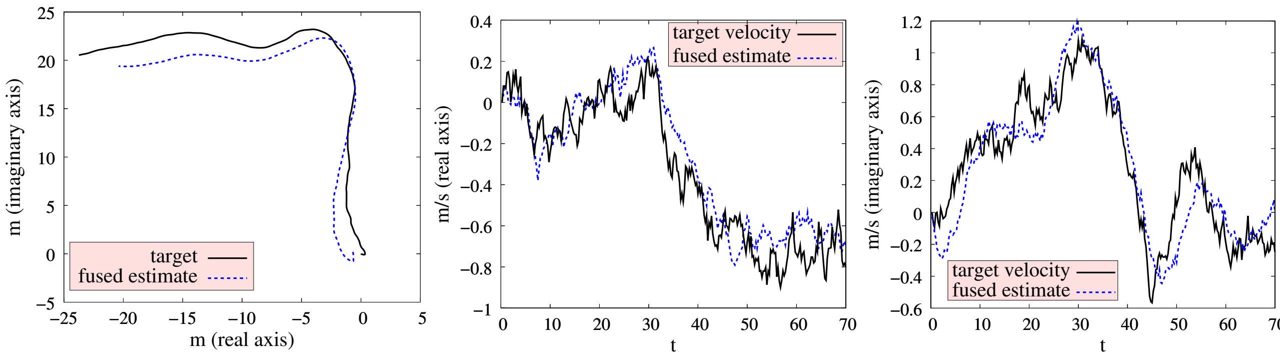

Fig. 1. Simulation parameters: $T_s = 0.25$ s, $N_c = 4$, $N_s = 20$, $M = 3000$ particle. **Left:** Estimate of the target trajectory during 70 s. **Middle:** Estimate of the target velocity on the real axis. **Right:** Estimate of the target velocity on the imaginary axis.

is the square centered at 0 with sides of length 100 m), $r_{max} = L$, $r_{min} = 10^{-4}$ and $c = 2$. The ARS procedure is iterated 3000 times for $\hat{r}^o_{2:4,0}$ and 1000 times for each $\hat{s}_{k,0}$, $k = 1, ..., N_s$.

The state prior is $p(\mathbf{x}_0) = CN\left(\mathbf{x}_0|[0, 0]^\top, \text{diag}\{10, 0.1\}\right)$ and the state noise is distributed as $u_t \sim CN(0, 0.1)$ (which is used for drawing the $\mathbf{x}_t^{(i)}$ particles in the APF). Finally, the fusion of the estimates is carried out by solving (4) via an ARS algorithm with random initial guess in $[0, 2\pi)$, $r_{max} = \pi$, $r_{min} = 10^{-5}$ and $c = 2$. The ARS procedure is iterated 1000 times.

Figure 1 shows the results of a typical simulation run with observation period $T_s = 0.25$ s, 70 s of total simulated time and $M = 3000$ particles in the APF. The left plot depicts the target trajectory on the complex plane and the track obtained by the fusion of the trajectory estimates from the $N_c = 4$ APFs. The middle and right plots show the fused estimates of the target velocity on the real and imaginary axis, respectively.

In order to estimate the average performance of the proposed method, we have carried out 40 independent computer simulations (each one with a different network deployment and target trajectory) and computed the mean absolute error (MAE) in the estimation of the target position, its velocity, the sensor locations and the DFC locations. The results are presented in Table 3 and illustrate the effectiveness of the method. Beware that the resulting global estimates may still be rotated with respect to the true trajectory and node locations. Although this problem is mitigated when a non-uniform prior pdf of the DFCs is available (as it is the case in this example), a further phase correction (applied to the global estimate) is necessary in order to obtain meaningful error values.

r_t	v_t	s_k	r^o_n
1.9602 m	0.3794 m/s	7.2284 m	3.1214 m

Table 3. Mean absolute error (MAE) in the estimation of the target position, r_t, given in m; the target velocity, v_t, in m/s; the position of a single sensor, s_k, $k \in \{1, ..., N_s\}$, in m; and the position of a single DFC, r^o_n, $n \in \{2, 3, 4\}$, in m.

5. CONCLUSIONS

We have proposed a Monte Carlo methodology that allows to jointly estimate the positions of the nodes of a sensor network (including both the sensors and the DFCs) and track a target that moves along the region monitored by the network. The method does not require the aid of beacons. Instead, a three-step procedure is carried out, that includes: (i) the acquisition of (rough) initial node location estimates by a random search optimization of the posterior probability of the node positions given a set of distance-aware measurements; (ii) target tracking and refinement of the node position estimates using a suitably designed auxiliary particle filter running independently at each DFC; and (iii) fusion (coherent combination) of the estimates provided by the different DFCs. We have presented computer simulation results that illustrate the successful application of the method with a network of power-aware sensors.

6. REFERENCES

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, “Locating the nodes,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.

[2] G. Sun, J. Chen, W. Guo, and K. J. R. Liu, “Signal processing techniques in network-aided positioning,” *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 12–23, July 2005.

[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation and tracking,” *IEEE Transactions Signal Processing*, vol. 50, no. 2, pp. 425–437, February 2002.

[4] J. Liu and M. West, “Combined parameter and state estimation in simulation-based filtering,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds., chapter 10, pp. 197–223. Springer, 2001.

[5] M. J. Appel, R. Labarre, and D. Radulovic, “On accelerated random search,” *SIAM Journal of Optimization*, vol. 14, no. 3, pp. 708–730, 2003.

[6] M. H. DeGroot and M. J. Schervish, *Probability and Statistics, 3rd ed.*, Addison-Wesley, New York, 2002.