COOPERATIVE TRACKING USING MARGINAL DIFFUSION PARTICLE FILTERS

Marcelo G. S. Bruno

Instituto Tecnológico de Aeronáutica
São José dos Campos SP 12228-900, Brazil
+55 12 3947-6906; bruno@ita.br

Stiven S. Dias

Embraer S.A.
São José dos Campos SP 12227-901, Brazil
+55 12 3927-5428; stiven.dias@embraer.com.br

ABSTRACT

This paper formulates the general Adapt-then-Combine (ATC) and Random Exchange (RndEx) diffusion filters for an arbitrary nonlinear state-space model. Subsequently, we propose two novel marginal Particle Filter implementations of the general ATC and RndEx filters using respectively a pure Sequential Monte Carlo (SMC) strategy and a hybrid Gaussian/SMC methodology. The proposed algorithms are assessed via simulation in a numerical example of cooperative target tracking with received-signal-strength (RSS) sensors.

Index Terms— Diffusion Filters, Marginal Particle Filters, RSS Target Tracking.

1. INTRODUCTION

In modern engineering systems, multiple agents dispersed over remote nodes of a network often cooperate with each other to execute a common signal processing task [1], [2] such as tracking a sequence of hidden state vectors. For linear, Gaussian state-space models, a well-known fully distributed solution to the cooperative signal tracking problem, where nodes have access to local measurements only, but exchange messages with their neighbors over partially connected networks, is the Adapt-then-Combine (ATC) linear diffusion Kalman filter introduced by Cattivelli and Sayed in [3].

More recently, the work by Dedecius and Djurić [4] has provided a more comprehensive Bayesian interpretation of ATC diffusion. In this paper, we use the general ATC Bayes filter formulation in [4] for cooperative tracking of a sequence of state vectors and propose two novel marginal particle filter [5] implementations of the general ATC Bayes filter in a scenario with nonlinear state-space models. The proposed implementations use respectively a pure Sequential Monte Carlo (SMC) methodology and a hybrid Gaussian / SMC strategy in the Adapt step, and a Gaussian approximation in the Combine step. The ATC diffusion filter is then compared to an alternative marginal Random Exchange (RndEx) diffusion particle filter, which is another novel algorithm introduced in this paper that uses the RndEx diffusion strategy originally described in [6], but, unlike the smoothing particle filter (PF) in [6], seeks to build a recursive Monte Carlo representation directly of the marginal posterior distribution of the unknown state at each instant n conditioned on a random subset of measurements coming from different locations in the network at different time instants.

The paper is divided into 7 sections. Sec. 1 is this Introduction. The general problem formulation is presented in Sec. 2. In Sec 3, we first review the general ATC diffusion Bayes filter described in [4] and then introduce the two proposed marginal PF implementations in Sec. 4. In Sec. 5, we discuss the alternative RndEx diffusion Bayes filter, which is formulated in a different way from [6], and describe again a marginal PF implementation of the RndEx filter for an arbitrary state-space model. Simulation results in an example of received-signal-strength (RSS) tracking are presented in Sec. 6. Finally, we present our conclusions in Sec. 7.

2. PROBLEM FORMULATION

Let $\{\mathbf{x}_n\}$, $n \geq 0$, be a sequence of hidden continuous random vectors[1] taking values in $\Re^N$. Likewise, let $\{\mathbf{y}_{n,r}\}$, $n \geq 0$, be a sequence of observed continuous random vectors taking values in $\Re^L$ such that the sequence $\{\mathbf{y}_{n,r}\}$ is available at the rth node of a network of R sensors. The sequences $\{\mathbf{x}_n\}$ and $\{\mathbf{y}_{n,r}\}$ are related, for $n \geq 0$ and $r \in \{1, 2, \ldots, R\}$, by the discrete-time state-space model

$$\mathbf{x}_{n+1} = \mathbf{f}_n(\mathbf{x}_n) + \mathbf{G}_n \mathbf{u}_n \quad (1)$$

$$\mathbf{y}_{n,r} = \mathbf{h}_{n,r}(\mathbf{x}_n) + \mathbf{v}_{n,r} \quad (2)$$

where the random vectors $\mathbf{x}_0$, $\{\mathbf{u}_n\}$, and $\{\mathbf{v}_{n,r}\}$ are mutually independent for all $n \geq 0$ and all $r \in \{1, 2, \ldots, R\}$; $\mathbf{u}_n$ and $\mathbf{v}_{n,r}$ are Gaussian with zero mean and covariance matrices respectively $\mathbf{Q}_n$ and $\mathbf{R}_{n,r}$, and $\mathbf{f}_n$ and $\mathbf{h}_{n,r}$ are arbitrary (possibly nonlinear) functions that are known for all $n \geq 0$ and all indexes $r \in \{1, 2, \ldots, R\}$. Our goal is to derive fully distributed algorithms that recursively estimate $\mathbf{x}_n$ at instant n given $\mathbf{y}_{0:n,1:R}$ defined as the vector that collects the observations $\{\mathbf{y}_{k,r}\}$ for $k = 0, 1, \ldots, n$ and $r = 1, 2, \ldots, R$.

[1] In this paper, we use lowercase letters, e.g. $\mathbf{x}$, to denote both random vectors and real vectors (including samples of random vectors), with the distinction implicit in context.

3. ATC DIFFUSION FILTER

The Adapt-then-Combine (ATC) diffusion filter originally described in a Bayesian formulation in [4] assumes that, at instant $n-1$, each network node r has an available posterior probability density function (p.d.f.) $p_{n-1|n-1,r}(\mathbf{x}_n)$ that actually depends on all network measurements that have contributed to its computation from instant zero up to instant $n-1$; for simplicity, we omit the conditioning on the measurements in the notation used in this Section. Furthermore, we use the symbol $\propto$ to denote "proportional to" with the implied normalization constant such that the function on the left-hand side of the expression integrates to one.

Adapt Step Let $\bar{N}(r)$ denote the *closed neighborhood* of the node r, which includes both $\{r\}$ and the indexes of all network nodes that are connected to node r according to the topology of the network graph. Each node r then first updates $p_{n-1|n-1,r}(\mathbf{x}_n)$ using the locally available observations $\{\mathbf{y}_{n,u}\}$, for all $u \in \bar{N}(r)$. Following the discussion in [4], under the mutual independence assumptions in Sec. 2, the general form of the Adapt step at node r at instant n is given by

$$\begin{aligned} p_{n|n-1,r}(\mathbf{x}_n) &= \int_{\Re^N} \big[\, p(\mathbf{x}_n|\mathbf{x}_{n-1}) \\ &\times \quad p_{n-1|n-1,r}(\,\mathbf{x}_{n-1}\,)\big]\; d\mathbf{x}_{n-1} \qquad (3) \\ \tilde{p}_{n|n,r}(\mathbf{x}_n) &\propto \left[\prod_{u\in\bar{N}(r)} p(\mathbf{y}_{n,u}|\mathbf{x}_n)\right] \\ &\times \quad p_{n|n-1,r}(\mathbf{x}_n). \qquad (4) \end{aligned}$$

Combine Step Following the local update in Eqs. (3) and (4), the Combine step replaces $\tilde{p}_{n|n,r}(\mathbf{x}_n)$ at each node r with a *merged* p.d.f.

$$p_{n|n,r}(\mathbf{x}_n) \propto \prod_{u\in\bar{N}(r)} \big[\tilde{p}_{n|n,u}(\mathbf{x}_n)\big]^{a_{r,u}} \qquad (5)$$

where $\sum_{u\in\bar{N}(r)} a_{r,u} = 1$, $\forall r$. As shown in [4], $p_{n|n,r}$ is the merged p.d.f. that minimizes, at instant n and at node r, the weighted average Kullback-Leibler (KL) divergence $\sum_{u\in\bar{N}(r)} a_{r,u} D_{KL}(p^\star\,||\tilde{p}_{n|n,u})$ over all possible p.d.f.'s $p^\star$.

4. ATC MARGINAL DIFFUSION PARTICLE FILTER

In the sequel, the notation $\mathbf{x} \sim p(\mathbf{x})$ indicates that $\mathbf{x}$ is a sample of a continuous random vector with p.d.f. p.

4.1. Marginal Particle Filter Adapt Step

Assume that node r at instant $n-1$ has a weighted set of samples $\{\mathbf{x}_{n-1,r}^{(j)}\}$, $j=1,\ldots,J$, with respective weights $\{w_{n-1,r}^{(j)}\}$, $\sum_j w_{n-1,r}^{(j)} = 1$, such that the aforementioned set of weighted samples represents the p.d.f. $p_{n-1|n-1,r}$ in the Monte Carlo sense. The goal of marginal particle filtering [5] in the context of this Section is to compute at instant n an updated weighted sample set $\{\tilde{\mathbf{x}}_{n,r}^{(j)}\}$ with respective weights $\{\tilde{w}_{n,r}^{(j)}\}$ that represents now the updated p.d.f. $\tilde{p}_{n|n,r}$ obtained from Eqs. (3) and (4). First, we make the Monte Carlo approximation $\int_{\Re^N} p(\mathbf{x}_n|\mathbf{x}_{n-1})\; p_{n-1|n-1,r}(\,\mathbf{x}_{n-1}\,)\; d\mathbf{x}_{n-1} \approx \sum_{l=1}^{J} w_{n-1,r}^{(l)}\, p(\mathbf{x}_n|\mathbf{x}_{n-1,r}^{(l)})$.

Next, using an importance sampling technique [8], [9] with a proposal p.d.f. $\pi_{n|n-1,r}$, a properly weighted set of particles to represent $\tilde{p}_{n|n,r}$, see also [5], is obtained by sampling $\tilde{\mathbf{x}}_{n,r}^{(j)} \sim \pi_{n|n-1,r}(\mathbf{x}_n)$ for $j=1,\ldots,J$ and then computing the corresponding updated weights

$$\tilde{w}_{n,r}^{(j)} \propto \left[\prod_{u\in\bar{N}(r)} p(\mathbf{y}_{n,u}|\tilde{\mathbf{x}}_{n,r}^{(j)})\right] \frac{\sum_{l=1}^{J} w_{n-1,r}^{(l)}\, p(\tilde{\mathbf{x}}_{n,r}^{(j)}|\mathbf{x}_{n-1,r}^{(l)})}{\pi_{n|n-1,r}(\tilde{\mathbf{x}}_{n,r}^{(j)})} \qquad (6)$$

for $j=1,\ldots,J$, where the proportionality constant in (6) is such that $\sum_{j=1}^{J} \tilde{w}_{n,r}^{(j)} = 1$. Let $\mathcal{N}(\mathbf{x}|\mathbf{m},\mathbf{P})$ denote a multivariate Gaussian p.d.f. with mean $\mathbf{m}$ and covariance matrix $\mathbf{P}$. A common choice for the proposal p.d.f., based on an Extended Kalman Filter (EKF) technique [10], is to make

$$\pi_{n|n-1,r}(\,\mathbf{x}_n\,) = \mathcal{N}(\mathbf{x}_n|\;\mathbf{m}_{n|n-1,r}, \boldsymbol{\Sigma}_{n|n-1,r}) \qquad (7)$$

where

$$\begin{aligned} \mathbf{m}_{n|n-1,r} &= \mathbf{f}_{n-1}(\hat{\mathbf{x}}_{n-1|n-1,r}) \qquad (8) \\ \boldsymbol{\Sigma}_{n|n-1,r} &= \tilde{\mathbf{F}}_{n-1}\,\mathbf{P}_{n-1|n-1,r}\,\tilde{\mathbf{F}}_{n-1}^T \\ &+ \mathbf{G}_{n-1}\mathbf{Q}_{n-1}\mathbf{G}_{n-1}^T \qquad (9) \end{aligned}$$

with $\hat{\mathbf{x}}_{n-1|n-1,r}$ and $\mathbf{P}_{n-1|n-1,r}$ being respectively the conditional mean and the conditional covariance matrix associated with $p_{n-1|n-1,r}$ at node r at instant $n-1$, and $\tilde{\mathbf{F}}_{n-1}$, as in the standard EKF algorithm when $\mathbf{f}_{n-1}(\mathbf{x}) = [f_{n-1,1}(\mathbf{x})\ldots f_{n-1,N}(\mathbf{x})]^T$, being the $N\times N$ matrix whose (i,j) element is the partial derivative $\frac{\partial f_{n-1,i}(\mathbf{x})}{\partial x_j}$ evaluated at $\mathbf{x}=\hat{\mathbf{x}}_{n-1|n-1,r}$.

4.2. Hybrid Gaussian/SMC Adapt Step

As shown in [5], the marginal particle filter Adapt step as described in Sec. 4.1 is asymptotically optimal as $J\to\infty$, but it has the inconvenience of complexity $\mathcal{O}(J^2)$. To reduce the computational complexity to $\mathcal{O}(J)$, we propose an alternative, *suboptimal* solution that is not limited to, but particularly useful when $\mathbf{f}_n(\mathbf{x}_n) = \mathbf{F}_n\,\mathbf{x}_n$, where $\mathbf{F}_n$ is a known $N\times N$ matrix for all $n\geq 0$. Under the latter condition, we define the approximate p.d.f. $\hat{p}_{n-1|n-1,r}(\mathbf{x}_{n-1})$ at instant $n-1$ at node r by $\mathcal{N}(\mathbf{x}_{n-1}|\hat{\mathbf{x}}_{n-1|n-1,r}, \mathbf{P}_{n-1|n-1,r})$ such that

$$\begin{aligned} \hat{p}_{n|n-1,r}(\mathbf{x}_n) &= \int p(\mathbf{x}_n|\mathbf{x}_{n-1})\hat{p}_{n-1|n-1,r}(\,\mathbf{x}_{n-1}\,)d\mathbf{x}_{n-1} \\ &= \mathcal{N}(\mathbf{x}_n|\,\hat{\mathbf{m}}_{n|n-1,r}, \hat{\boldsymbol{\Sigma}}_{n|n-1,r}) \end{aligned}$$

with $\hat{\mathbf{m}}_{n|n-1,r}$ and $\hat{\mathbf{\Sigma}}_{n|n-1,r}$ given by Eqs. (8) and (9) replacing, however, $\mathbf{f}_{n-1}(\hat{\mathbf{x}}_{n-1|n-1,r})$ with $\mathbf{F}_{n-1}\,\hat{\mathbf{x}}_{n-1|n-1,r}$ and $\tilde{\mathbf{F}}_{n-1}$ with $\mathbf{F}_{n-1}$. In the sequel, for $j = 1, \ldots, J$

1. Draw $\tilde{\mathbf{x}}_{n|n,r}^{(j)} \sim \hat{p}_{n|n-1,r}(\mathbf{x}_n)$.

2. Update the weights by making

$$\tilde{w}_{n,r}^{(j)} \propto \left[\prod_{u \in \bar{N}(r)} p(\mathbf{y}_{n,u}|\tilde{\mathbf{x}}_{n,r}^{(j)}) \right].$$

4.3. Combine Step

In order to merge the local posterior p.d.f's in the Combine step, a parametric approximation to $\tilde{p}_{n|n,r}$ is needed at each node r at instant n. The simplest approximation is to use the particle set $\{\tilde{\mathbf{x}}_{n|n,r}^{(j)}\}$ with respective weights $\{\tilde{w}_{n|n,r}^{(j)}\}$ to compute the sample mean vector and the sample covariance matrix at node r and instant n given by $\tilde{\mathbf{x}}_{n|n,r} = \sum_{l=1}^{J} \tilde{w}_{n,r}^{(l)}\, \tilde{\mathbf{x}}_{n|n,r}^{(l)}$ and $\tilde{\mathbf{P}}_{n|n,r} = \sum_{l=1}^{J} \tilde{w}_{n,r}^{(l)}\, (\mathbf{x}_{n|n,r}^{(l)} - \tilde{\mathbf{x}}_{n|n,r})\, (\mathbf{x}_{n|n,r}^{(l)} - \tilde{\mathbf{x}}_{n|n,r})^T$.

In the sequel, we approximate $\tilde{p}_{n|n,r}(\mathbf{x}_n) \approx \mathcal{N}(\mathbf{x}_n|\,\tilde{\mathbf{x}}_{n|n,\,r}, \tilde{\mathbf{P}}_{n|n,\,r})$. Under the Gaussian approximation, the merged p.d.f. $p_{n|n,r}$ on the left-hand side of the KL fusion rule in (5) is also Gaussian, see [4] and [11], with fused covariance matrix and mean vector given by

$$\begin{aligned} (\mathbf{P}_{n|n,r})^{-1} &= \sum_{u \in \bar{N}(r)} a_{r,u} (\tilde{\mathbf{P}}_{n|n,u})^{-1} \\ \hat{\mathbf{x}}_{n|n,r} &= \mathbf{P}_{n|n,r} \left[\sum_{u \in \bar{N}(r)} a_{r,u}\, (\tilde{\mathbf{P}}_{n|n,u})^{-1}\, \tilde{\mathbf{x}}_{n|n,u} \right]. \end{aligned}$$

Finally, node r resamples $\mathbf{x}_{n\,r}^{(j)} \sim \mathcal{N}(\mathbf{x}_n|\,\hat{\mathbf{x}}_{n|n,\,r}, \mathbf{P}_{n|n,\,r})$ and resets $w_{n,r}^{(j)} = 1/J$ for $j = 1, 2, \ldots, J$.

5. MARGINAL RNDEX DIFFUSION FILTER

The alternative Random Exchange (RndEx) diffusion methodology assumes that, at instant $n-1$, a given network node s, stores a posterior p.d.f. $p(\mathbf{x}_{n-1}|\tilde{\mathbf{y}}_{0:n-1,s})$ where $\tilde{\mathbf{y}}_{0:n-1,s}$ denotes a random subset of $\mathbf{y}_{0:n-1,1:R}$. The network then executes the fully-distributed asynchronous random exchange (RndEx) protocol described in detail in [6] and [7], after which the p.d.f. $p(\mathbf{x}_{n-1}|\tilde{\mathbf{y}}_{0:n-1,s})$ (or rather a parametric representation thereof) ends up at another random node r, which is not necessarily in the neighborhood of s. Next, at instant n, node r updates its new stored p.d.f. using the locally available observations in the vector $\tilde{\mathbf{y}}_{n,r}$ that collects the measurements $\{\mathbf{y}_{n,u}\}$ for all $u \in \bar{N}(r)$, i.e. node r computes at instant n

$$\begin{aligned} p(\mathbf{x}_n|\tilde{\mathbf{y}}_{0:n-1,s}) &= \int_{\Re^N} [\, p(\mathbf{x}_n|\mathbf{x}_{n-1}) \\ &\quad \times \; p(\mathbf{x}_{n-1}|\tilde{\mathbf{y}}_{0:n-1,s})\,]\, d\mathbf{x}_{n-1} \\ p(\mathbf{x}_n|\tilde{\mathbf{y}}_{n,r}, \tilde{\mathbf{y}}_{0:n-1,s}) &\propto \left[\prod_{u \in \bar{N}(r)} p(\mathbf{y}_{n,u}|\mathbf{x}_n) \right] \\ &\quad \times \; p(\mathbf{x}_n|\tilde{\mathbf{y}}_{0:n-1,s}). \end{aligned}$$

We redefine now $\tilde{\mathbf{y}}_{0:n,r} = \left[\tilde{\mathbf{y}}_{0:n-1,s}^T\, \tilde{\mathbf{y}}_{n,r}^T\right]^T$. As a given p.d.f. travels through a random path of nodes $\{l_0, l_1, \ldots, l_n\}$ over the network between instants zero and n, node l_n at instant n, after local data assimilation, stores then the posterior p.d.f. $p(\mathbf{x}_n|\tilde{\mathbf{y}}_{0,l_0}\, \tilde{\mathbf{y}}_{1,l_1}\, \ldots\, \tilde{\mathbf{y}}_{n,l_n})$, thus enabling data diffusion.

If $p(\mathbf{x}_{n-1}|\tilde{\mathbf{y}}_{0:n-1,s})$ is represented at instant $n-1$ by a set of weighted samples $\{\mathbf{x}_{n-1,s}^{(j)}\}$, $j = 1, \ldots, J$, with weights $\{w_{n-1,s}^{(j)}\}$, we may approximate $p(\mathbf{x}_{n-1}|\tilde{\mathbf{y}}_{0:n-1,s})$ by $\mathcal{N}(\mathbf{x}_{n-1}|\,\hat{\mathbf{x}}_{n-1|n-1,\,s},\, \mathbf{P}_{n-1|n-1,\,s})$ where $\hat{\mathbf{x}}_{n-1|n-1,\,s}$ and $\mathbf{P}_{n-1|n-1,\,s}$ are respectively the sample mean and sample covariance matrix computed from the weighted set $\{(w_{n-1,s}^{(j)}, \mathbf{x}_{n-1,s}^{(j)})\}$. Upon receiving $\hat{\mathbf{x}}_{n-1|n-1,\,s}$ and $\mathbf{P}_{n-1|n-1,\,s}$ from node s during the random exchange step, node r at instant n builds a new Monte Carlo representation for $p(\mathbf{x}_n|\tilde{\mathbf{y}}_{0:n,r})$ using the following marginal PF algorithm:

For $j = 1, \ldots, J$

1. Resample $\mathbf{x}_{n-1,s}^{(j)} \sim \mathcal{N}(\mathbf{x}_{n-1}|\,\hat{\mathbf{x}}_{n-1|n-1\,s}, \mathbf{P}_{n-1|n-1,\,s})$ and set $w_{n-1,s}^{(j)} = \frac{1}{J}$.

2. Sample $\mathbf{x}_{n,r}^{(j)} \sim \pi_{n|n-1,s}(\mathbf{x}_n)$, where $\pi_{n|n-1,s}(\mathbf{x}_n)$ is defined as in Eqs. (7), (8) and (9) replacing $\hat{\mathbf{x}}_{n-1|n-1,\,r}$ and $\mathbf{P}_{n-1|n-1,\,r}$ with $\hat{\mathbf{x}}_{n-1|n-1,\,s}$ and $\mathbf{P}_{n-1|n-1,\,s}$.

3. Update the sample weights as

$$\begin{aligned} w_{n,r}^{(j)} &\propto \left[\prod_{u \in \bar{N}(r)} p(\mathbf{y}_{n,u}|\tilde{\mathbf{x}}_{n,r}^{(j)}) \right] \\ &\quad \times \; \frac{\sum_{l=1}^{J} w_{n-1,s}^{(l)}\, p(\tilde{\mathbf{x}}_{n,r}^{(j)}|\mathbf{x}_{n-1,s}^{(l)})}{\pi_{n|n-1,s}(\tilde{\mathbf{x}}_{n,r}^{(j)})}. \end{aligned} \quad (10)$$

Again, to eliminate the quadratic complexity of the marginal PF algorithm in the number of particles, a simplified hybrid Gaussian/SMC implementation as in Sec. 4.2 can be used, especially when the state model is linear.

Remark Note that the smoothing RndEx diffusion PF in [6], at the end of the random node path $\{l_0, l_1, \ldots, l_n\}$, builds in theory at node l_n a Monte Carlo representation of $p(\mathbf{x}_{0:n}|\tilde{\mathbf{y}}_{0,l_0}\, \tilde{\mathbf{y}}_{1,l_1}\, \ldots\, \tilde{\mathbf{y}}_{n,l_n})$, rather than $p(\mathbf{x}_n|\tilde{\mathbf{y}}_{0,l_0}\, \tilde{\mathbf{y}}_{1,l_1}\, \ldots\, \tilde{\mathbf{y}}_{n,l_n})$ directly, and, therefore, is both algorithmically and conceptually different from the marginal RndEx PF formulated in this paper.

6. SIMULATION EXAMPLE

As an illustrative example, we track the sequence of hidden state vectors $\{\mathbf{x}_n\}$ that collect the positions and velocities of a moving target's centroid in a two-dimensional (2-D) surveillance space of size approximately 120×120 meters (m). The random sequence $\{\mathbf{x}_n\}$ evolves in time according to the linear, white-noise acceleration model

$$\mathbf{x}_{n+1} = \mathbf{F}\,\mathbf{x}_n + \mathbf{u}_n$$

described in [12], where the block-diagonal matrices $\mathbf{F}$ and $\mathbf{Q}$ are given by $\mathbf{F} = diag(\tilde{\mathbf{F}}, \tilde{\mathbf{F}})$ and $\mathbf{Q} = diag(\tilde{\mathbf{Q}}, \tilde{\mathbf{Q}})$ with $\tilde{\mathbf{F}} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $\tilde{\mathbf{Q}} = \sigma^2_{accel} \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix}$, and T is the sampling period. Detailed parameters for the simulated scenario are provided in [6].

A partially connected network of 25 received-signal-strength (RSS) sensors records at each instant n the measurements $\{y_{n,r}\}$ in dBm at each network location r such that [13]

$$y_{n,r} = \underbrace{P_0 - 10\zeta_r \log_{10}\left(\frac{||\mathbf{H}\,\mathbf{x}_n - \mathbf{x}_r||}{d_0}\right)}_{h_{n,r}(\mathbf{x}_n)} + v_{n,r} \qquad (11)$$

where $\mathbf{x}_r$ is the (known) r-th sensor position in the 2-D surveillance space, $||.||$ is the Euclidean norm in $\Re^2$; (P_0, d_0, ζ_r) are known model parameters (see [13] for details), and $\mathbf{H}$ is a 2×4 projection matrix such that $H(1,1) = H(2,3) = 1$ and $H(i,j) = 0$ otherwise.

We ran marginal PF implementations respectively of the optimal (centralized) network filter, the RndEx diffusion filter, and the ATC diffusion filter. For the ATC diffusion filter in particular, we also implemented an iterative version the Combine step in Sec. 4.3 with 10 iterations. As discussed in a different context in [14], the iterative version of Eq. (5) converges, as the number of iterations go to infinity, to a fused p.d.f. that minimizes over all possible p.d.f.'s $p^\star$ the average network KL divergence $\sum_{u=1}^{R} \left[\frac{1}{R}\, D_{KL}(p^\star\,||\tilde{p}_{n|n,u})\right]$.

All marginal PFs in our simulations used a simplified hybrid Gaussian/SMC implementation with $J = 1000$ particles at each Monte Carlo run. Fig. 1 shows the empirical root-mean-square (RMS) position estimate error averaged over all network nodes from instant zero up to instant 100 s obtained from Monte Carlo simulations for each tested algorithm. Table 1 on the other hand compares the evaluated algorithms in terms of their processing and communication costs, which were computed according to the methodology detailed in [15]

We see from Fig. 1 that all tested filters tracked the emitter with small steady-state error. The ATC filter with an iterative Combine step was the closest in performance to the optimal centralized filter in this example, but, as shown in Table 1, its associated internode communication cost in our simulations was about one order of magnitude higher than those for the RndEx and the non-iterative ATC diffusion filters. The non-iterative ATC filter, however, showed a slight improvement in RMSE over the RndEx-MPF, with an internode communication cost that is only about twice as high as that of the latter. Note that both non-iterative ATC-MPF and RndEx-MPF are scalable with network size since their internode communication cost depends on the network degree only.

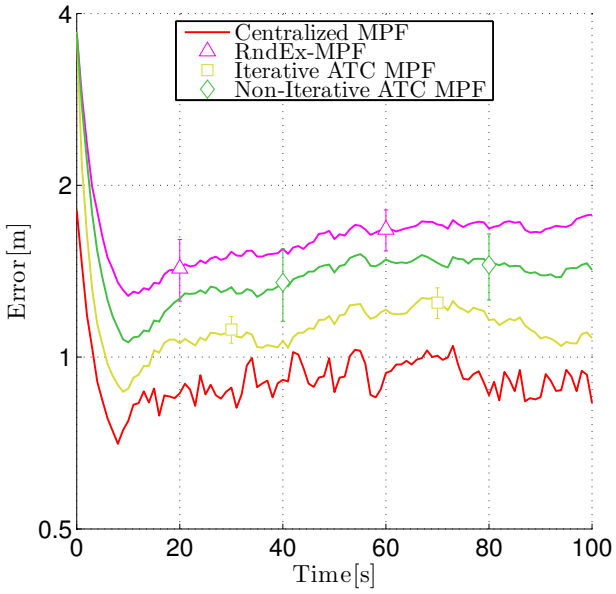

Fig. 1. Evolution of the estimated position RMS error norm.

Table 1. Communication and processing performances.

Evaluated Algorithm	RX Rate	TX Rate	Duty Cycle
RndEx-MPF	148 B/s	132 B/s	2.7 %
Iterative ATC MPF	2.9 KB/s	604 B/s	2.8 %
Non-Iterative ATC MPF	317 B/s	64 B/s	2.4 %

7. CONCLUSIONS

We presented in this paper the general Bayesian formulation of the fully distributed ATC and RndEx diffusion filters for cooperative signal tracking over sensor networks. We then derived novel marginal particle filter implementations of the ATC and RndEx filter using a combination of sequential Monte Carlo methods and Gaussian parametric approximations; alternatively, Gaussian sum approximations could also have been used and are left for future work. In particular, using the hybrid SMC/Gaussian approach, we managed to reduce the complexity of the marginal PF from $\mathcal{O}(J^2)$ to $\mathcal{O}(J)$, where J is the number of particles. The proposed filters were tested in a numerical example with simulated RSS sensor data. The simulations show that, overall, all tested filters performed well in terms of state estimation accuracy, but the non-iterative marginal ATC particle filter offered the best compromise between internode communication cost and RMSE performance.

8. REFERENCES

[1] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed Particle Filtering in Agent Networks: A Survey, Classification, and Comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, January 2013.

[2] A. H. Sayed, S. Y. Tu, J. Chen, and X. Zhao, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.

[3] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, September 2010.

[4] K. Dedecius, and P. M. Djurić, "Sequential estimation and diffusion of information over networks: A Bayesian approach with exponential family of distributions," *IEEE Transactions on Signal Processing* vol. 65, no. 7, pp. 1705-1809, April 2017.

[5] M. Klaas, N. De Freitas and A. Doucet, "Toward practical N2 Monte Carlo: The marginal particle filter," in *Proc. 21th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, AUAI Press, pp. 308–315, Arlington, Virginia, 2005.

[6] M. G. S. Bruno and S. S. Dias, "Cooperative emitter tracking using Rao-Blackwellized random exchange diffusion particle filters," *Eurasip Journal on Advances in Signal Processing*, 2014:19, pp. 1–18, February 2014.

[7] S . S. Dias and M. G. S. Bruno, "Distributed Bernoulli filters for joint detection and tracking in sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 3, pp. 260–275, September 2016.

[8] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 3, pp. 197–208, 2000.

[9] M. G. S. Bruno, "Sequential Monte Carlo methods for nonlinear discrete-time filtering," *Synthesis Lectures on Signal Processing*, J. M. F. Moura, Editor, vol. 6, pp. 1–99, Morgan & Claypool Publishers, 2013.

[10] B. 0. Anderson and J. B. Moore, "Optimal Filtering," *Information and Systems Science Series*, T. Kailath Editor, Prentice-Hall Inc., Englewood Cliffs, N.J. U.S.A., pp. 1–357, 1979.

[11] I. Hu, L. Xie, and C. Zhang, "Diffusion Kalman filtering based on covariance intersection," *IEEE Transactions on Signal Processing*", vol. 60, no. 2, pp. 891–902, February 2012.

[12] Y. Bar-Shalom and X. Li, "Multitarget-Multisensor Tracking: Principles and Techniques," YBS, Storrs, CT, 1995.

[13] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 2137-2148, March 2003.

[14] L. Chisci, G. Battistelli, C. Fantacci, A. Farina, and B. N. Vo, "Average Kullback-Leibler divergence for random finite sets," *Proc. 18th Int'l Conf. on Information Fusion*, Washington, D.C., July 6-9, 2015.

[15] S. S. Dias, and M. G. S. Bruno, "Distributed Emitter Tracking using Random Exchange Diffusion Particle Filters," *16th International Conference on Information Fusion*, Istanbul, Turkey, July 2013.