

Time-space-sequential Distributed Particle Filtering with Low-rate Communications

Ondrej Hlinka^a, Petar M. Djurić^b, and Franz Hlawatsch^a

^aInstitute of Communications and Radio-Frequency Engineering, Vienna University of Technology (ohlinka@nt.tuwien.ac.at)

^bDepartment of Electrical and Computer Engineering, Stony Brook University (djuric@ece.sunysb.edu)

Abstract—We present a distributed particle filtering scheme for time-space-sequential Bayesian state estimation in wireless sensor networks. Low-rate inter-sensor communications between neighboring sensors are achieved by transmitting Gaussian mixture (GM) representations instead of particles. The GM representations are calculated using a clustering algorithm. We also propose a “look-ahead” technique for designing the proposal density used for importance sampling. Simulation results for a target tracking application demonstrate the performance of our distributed particle filter and, specifically, the advantage of the look-ahead proposal design over a conventional design.

Index Terms—Wireless sensor network, distributed particle filter, Bayesian state estimation, Gaussian mixture, target tracking.

I. INTRODUCTION

Wireless sensor networks (WSNs) using distributed inference techniques are advantageous due to their scalability, flexibility in deployment, and robustness [1]. Because of the high energy consumption of inter-sensor wireless communications, it is often beneficial to limit communications to neighboring sensors only and to trade off communication for computation.

Here, we consider distributed, sequential, Bayesian state estimation in WSNs with low-rate communications between neighboring sensors. For the general nonlinear/non-Gaussian case, a computationally feasible approximation to sequential Bayesian state estimation is provided by the particle filter (PF) [2], [3]. However, because of the high communication cost of transmitting the particles, standard PF algorithms may not be well suited for distributed estimation in WSNs. In [4], a distributed PF with low-rate transmissions of Gaussian mixture (GM) representations was proposed. The GM representations are calculated from particle representations by means of the expectation-maximization (EM) algorithm.

In this paper, we extend [4] and [5] by proposing a distributed PF with low-rate communications and improved estimation performance. The proposed estimator is sequential both in time and in space (i.e., across sensors); it extends our estimator in [5] by using PFs, GMs, and improved sensor scheduling. Similarly to [4], we use GMs as low-rate approximations of posterior densities. However, these GMs are calculated by k-means clustering [6] rather than by the (more complex) EM

algorithm. Furthermore, we propose a “look-ahead” scheme that reduces the estimation error resulting from the GM approximations. This scheme uses a back-transmitted measurement of the next sensor and a suitably designed proposal density that is calculated by a Gaussian sum filter [7].

We note that related distributed estimators were proposed in [8], [9]; however, they do not use PFs and are not sequential in both time and space. Furthermore, the distributed PF methods proposed in [10], [11] do not use GM approximations.

This paper is organized as follows. In Section II, we review the time-space-sequential estimation framework of [5]. In Section III, we discuss the PF/GM computations performed at the sensors. Section IV presents the look-ahead scheme. In Section V, a target tracking application is considered and a decentralized sensor scheduling algorithm is proposed. Simulation results are presented in Section VI.

II. TIME-SPACE-SEQUENTIAL DISTRIBUTED ESTIMATION

State-space model. We consider a generally nonlinear and non-Gaussian dynamic system. The random state vector \mathbf{x}_n evolves with discrete time n according to the state-transition equation

$$\mathbf{x}_n = \mathbf{g}_n(\mathbf{x}_{n-1}, \mathbf{u}_n), \quad n = 1, 2, \dots, \quad (1)$$

where \mathbf{u}_n is white driving noise with a known probability density function (pdf) $f(\mathbf{u}_n)$. At time n , the state \mathbf{x}_n is sensed by K_n sensors according to the measurement equations

$$\mathbf{z}_n^k = \mathbf{h}_n^k(\mathbf{x}_n, \mathbf{w}_n^k), \quad k = 1, 2, \dots, K_n, \quad (2)$$

where \mathbf{z}_n^k is the measurement vector at time n and sensor k , and \mathbf{w}_n^k is measurement noise with known pdf $f(\mathbf{w}_n^k)$ and with $\mathbf{w}_n^k, \mathbf{w}_n^{k'}$ independent unless both $n = n'$ and $k = k'$. The initial state \mathbf{x}_0 and the sequences \mathbf{u}_n and \mathbf{w}_n^k are all independent. Hereafter, $\mathbf{z}_n^{1:k} \triangleq (\mathbf{z}_n^{1\top} \dots \mathbf{z}_n^{k\top})^\top$ will denote the current measurements of all sensors up to sensor k , with $\mathbf{z}_n^{1:0}$ being the empty vector. Furthermore, we also define the vector of all measurements at time n , $\mathbf{z}_n \triangleq \mathbf{z}_n^{1:K_n}$, and the total measurement vector up to time n , $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^\top \dots \mathbf{z}_n^\top)^\top$. Equations (1) and (2) together with our statistical assumptions determine the *state-transition pdf* $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ and the *likelihood functions* $f(\mathbf{z}_n^k|\mathbf{x}_n)$ and $f(\mathbf{z}_n|\mathbf{x}_n)$, respectively.

In the Bayesian setting, estimating the state \mathbf{x}_n from $\mathbf{z}_{1:n}$ essentially amounts to calculating the *posterior pdf* $f(\mathbf{x}_n|\mathbf{z}_{1:n})$.

This work was supported by the FWF under Award S10603-N13 within the National Research Network SISE. P. M. D. was supported by the NSF under Award CCF-0515246 and by the ONR under Award N00014-09-1-1154.

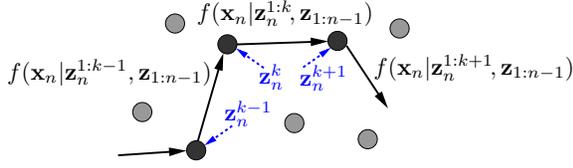


Figure 1. Space-sequential estimation at time n : The PP is transmitted from one sensor to the next and updated at each sensor using the local measurement \mathbf{z}_n^k .

A time-space sequential, distributed scheme for calculating $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ has been presented in [5]. This scheme, reviewed in the following, forms the basis for the distributed PF to be proposed in Section III.

Time recursion. As is well known [12], the current posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ can be calculated time-sequentially from the old posterior $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ and the current measurements \mathbf{z}_n . First, the “predicted posterior” $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$ is calculated from the old posterior $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ according to

$$f(\mathbf{x}_n|\mathbf{z}_{1:n-1}) = \int f(\mathbf{x}_n|\mathbf{x}_{n-1})f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})d\mathbf{x}_{n-1}. \quad (3)$$

Then, the current posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is obtained by updating $f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$, using the current measurements \mathbf{z}_n :

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) = \frac{f(\mathbf{z}_n|\mathbf{x}_n)f(\mathbf{x}_n|\mathbf{z}_{1:n-1})}{\int f(\mathbf{z}_n|\mathbf{x}_n)f(\mathbf{x}_n|\mathbf{z}_{1:n-1})d\mathbf{x}_n}. \quad (4)$$

Space recursion. Recall that at time n , the state \mathbf{x}_n is sensed by K_n sensors indexed as $k = 1, \dots, K_n$. The time update (4) involves the measurements at *all* sensors (vector \mathbf{z}_n). However, (4) can be reformulated in a *space-sequential, distributed* manner as follows (see Fig. 1). Sensor k receives from the previous sensor $k-1$ the “partial posterior” (PP) $f(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ that incorporates all past measurements $\mathbf{z}_{1:n-1}$ and all current measurements up to sensor $k-1$, $\mathbf{z}_n^{1:k-1}$. Next, sensor k calculates the updated PP $f(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ that incorporates also the local measurement \mathbf{z}_n^k :

$$f(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1}) = \frac{f(\mathbf{z}_n^k|\mathbf{x}_n)f(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})}{\int f(\mathbf{z}_n^k|\mathbf{x}_n)f(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})d\mathbf{x}_n}. \quad (5)$$

The updated PP is then transmitted to the next sensor $k+1$, where the same type of update is performed, etc. At the last sensor ($k = K_n$), the “total posterior” $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is obtained since $f(\mathbf{x}_n|\mathbf{z}_n^{1:K_n}, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n|\mathbf{z}_n, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n|\mathbf{z}_{1:n})$. The sequence of space recursions is initialized at sensor $k=1$ by $f(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})|_{k=1} = f(\mathbf{x}_n|\mathbf{z}_n^{1:0}, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n|\mathbf{z}_{1:n-1})$. This is the predicted posterior, which was calculated previously according to (3). We note that this space-sequential update was presented before, e.g., in [8], however without considering a temporal state transition.

The space-sequential formulation (5) of the time update (4) yields a *time-space-sequential algorithm* for calculating the temporal sequence of posteriors $f(\mathbf{x}_n|\mathbf{z}_{1:n})$, $n = 1, 2, \dots$. This algorithm is fully distributed and uses only communication between neighboring sensors.

III. DISTRIBUTED PARTICLE FILTER WITH GAUSSIAN MIXTURE TRANSMISSIONS

In the nonlinear/non-Gaussian case, the PF methodology can be used for an approximate calculation of (3) and (5). However, to obtain low-rate inter-sensor communications, we transmit GM representations of PPs (GMPPs), instead of particles, between neighboring sensors [4]. At sensor k , the PP $f(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ is approximated by the GMPP

$$f_{\text{GM}}(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1}) \triangleq \sum_{m=1}^M \alpha_{n,m}^k \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n,m}^k, \mathbf{C}_{n,m}^k), \quad (6)$$

where the $\alpha_{n,m}^k$ are nonnegative mixing coefficients satisfying $\sum_{m=1}^M \alpha_{n,m}^k = 1$ and $\mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{n,m}^k, \mathbf{C}_{n,m}^k)$ denotes a multivariate Gaussian pdf with mean $\boldsymbol{\mu}_{n,m}^k$ and covariance $\mathbf{C}_{n,m}^k$.

Next, we describe the processing performed at the sensors.

A. Update of partial posteriors

The PP update at sensor k (with some modifications for $k=1$, see below) proceeds as follows. Sensor k receives from sensor $k-1$ the parameters $\{\alpha_{n,m}^{k-1}, \boldsymbol{\mu}_{n,m}^{k-1}, \mathbf{C}_{n,m}^{k-1}\}_{m=1}^M$ of the previous GMPP $f_{\text{GM}}(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$. Sensor k then calculates a *particle representation* of the current PP $f(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ by means of importance sampling. First, S particles $\mathbf{x}_n^{(i)}$, $i = 1, \dots, S$ are drawn from a proposal pdf $g(\mathbf{x}_n)$. For each particle $\mathbf{x}_n^{(i)}$, an importance weight $\tilde{w}_n^{(i)}$ is calculated as (cf. (5))

$$\tilde{w}_n^{(i)} = \frac{f(\mathbf{z}_n^k|\mathbf{x}_n^{(i)})f_{\text{GM}}(\mathbf{x}_n^{(i)}|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})}{g(\mathbf{x}_n^{(i)})}, \quad (7)$$

and the $\tilde{w}_n^{(i)}$ are normalized as $w_n^{(i)} = \tilde{w}_n^{(i)} / \sum_{j=1}^S \tilde{w}_n^{(j)}$. The proposal pdf can be chosen as the received GMPP, i.e., $g(\mathbf{x}_n) = f_{\text{GM}}(\mathbf{x}_n|\mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$, as suggested in [4]. Improved proposal designs will be presented in Section IV.

From the particle representation $\{\mathbf{x}_n^{(i)}, w_n^{(i)}\}_{i=1}^S$ of $f(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$, a GMPP $f_{\text{GM}}(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ is now constructed as follows. The particles $\mathbf{x}_n^{(i)}$ are partitioned into M clusters by means of *k-means clustering* [6]. Let \mathcal{I}_m denote the set of indices i of all particles $\mathbf{x}_n^{(i)}$ assigned to cluster m . For each cluster $m \in \{1, \dots, M\}$, we calculate the sample mean $\boldsymbol{\mu}_{n,m}^k \triangleq \sum_{i \in \mathcal{I}_m} w_n^{(i)} \mathbf{x}_n^{(i)}$, sample covariance $\mathbf{C}_{n,m}^k \triangleq \sum_{i \in \mathcal{I}_m} w_n^{(i)} (\mathbf{x}_n^{(i)} - \boldsymbol{\mu}_{n,m}^k)(\mathbf{x}_n^{(i)} - \boldsymbol{\mu}_{n,m}^k)^\top$, and mixing coefficient $\alpha_{n,m}^k \triangleq \sum_{i \in \mathcal{I}_m} w_n^{(i)}$. The parameters $\alpha_{n,m}^k, \boldsymbol{\mu}_{n,m}^k, \mathbf{C}_{n,m}^k$ define the m th component of the GMPP $f_{\text{GM}}(\mathbf{x}_n|\mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ according to (6); they are transmitted (for all $m = 1, \dots, M$) to sensor $k+1$.

B. Update and prediction steps at last/first sensor

The last sensor at time $n-1$, $k = K_{n-1}$, receives from the previous sensor $K_{n-1}-1$ the parameters of the GMPP $f_{\text{GM}}(\mathbf{x}_{n-1}|\mathbf{z}_{n-1}^{1:K_{n-1}-1}, \mathbf{z}_{1:n-2})$. It then performs its update step by calculating a particle representation $\{\mathbf{x}_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^S$ of the total posterior at time $n-1$, $f(\mathbf{x}_{n-1}|\mathbf{z}_{n-1}^{1:K_{n-1}}, \mathbf{z}_{1:n-2}) = f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$, as described in Section III-A.

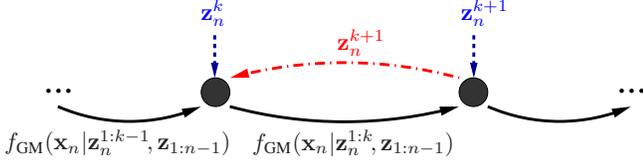


Figure 2. Look-ahead scheme: Sensor k uses the next measurement \mathbf{z}_n^{k+1} (back-transmitted from sensor $k+1$) to construct a better GM approximation $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$, which is then transmitted to sensor $k+1$.

For convenience, we assume that this last sensor at time $n-1$ is also the first sensor at time n , $k=1$. This first sensor uses a *local PF* to perform the time prediction step (3) as well as the measurement update step (cf. (5)) $f(\mathbf{x}_n | \mathbf{z}_n^{1:1}, \mathbf{z}_{1:n-1}) \propto f(\mathbf{z}_n^1 | \mathbf{x}_n) f(\mathbf{x}_n | \mathbf{z}_{1:n-1})$ (this update step incorporates the local measurement \mathbf{z}_n^1 ; note that $\mathbf{z}_n^{1:1} = \mathbf{z}_n^1$). The local PF works as follows. First, the *resampling* [2], [3] of $\{\mathbf{x}_{n-1}^{(i)}, w_{n-1}^{(i)}\}_{i=1}^S$ produces S resampled particles $\bar{\mathbf{x}}_{n-1}^{(i)}$ and identical weights $\bar{w}_{n-1}^{(i)} \equiv 1/S$. Next, for each $\bar{\mathbf{x}}_{n-1}^{(i)}$, a new particle $\mathbf{x}_n^{(i)}$ is drawn from $f(\mathbf{x}_n | \bar{\mathbf{x}}_{n-1}^{(i)})$ (prediction step), and the associated importance weight is calculated according to

$$w_n^{(i)} = \frac{f(\mathbf{z}_n^1 | \mathbf{x}_n^{(i)})}{\sum_{j=1}^S f(\mathbf{z}_n^1 | \mathbf{x}_n^{(j)})}$$

(update step). Finally, the parameters of the GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:1}, \mathbf{z}_{1:n-1})$ are obtained by means of clustering and transmitted to the next sensor ($k=2$). We note that this local PF is modified in the look-ahead scheme, as explained in Section IV-A.

IV. LOOK-AHEAD PROPOSAL DESIGN

We now present a variant of the method described above that achieves a performance improvement by using a modified proposal pdf for importance sampling.

A. Look-ahead scheme

In the improved method, the measurement \mathbf{z}_n^{k+1} of the next sensor $k+1$ is transmitted back to the currently active sensor k , as visualized in Fig. 2. The sensor k uses \mathbf{z}_n^{k+1} to calculate an improved GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ that is better suited to the processing performed at the sensor $k+1$ (“look-ahead”). This is achieved by using in the weight calculation (7) a proposal pdf $g(\mathbf{x}_n)$ that depends not only on the received GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ but also on the local measurement \mathbf{z}_n^k and the back-transmitted measurement \mathbf{z}_n^{k+1} . The construction of $g(\mathbf{x}_n)$ will be explained in Section IV-B. The underlying principle is that $g(\mathbf{x}_n)$ is similar to the next PP $f(\mathbf{x}_n | \mathbf{z}_n^{1:k+1}, \mathbf{z}_{1:n-1})$, so that the particles $\mathbf{x}_n^{(i)}$ drawn from $g(\mathbf{x}_n)$ are located in regions where the next PP is concentrated. From these particles, the current GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$ is constructed as explained in Section III-A. It follows that *this GMPP will approximate the current PP more accurately in regions relevant to the next PP*. As we will show in Section VI, this results in a performance improvement.

The processing at the first sensor ($k=1$) is also modified in the look-ahead scheme. After drawing particles $\mathbf{x}_n^{(i)}$ from the pdf’s $f(\mathbf{x}_n | \bar{\mathbf{x}}_{n-1}^{(i)})$ as described in Section III-B, clustering is performed to obtain a GM representation $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_{1:n-1})$ of the predicted posterior $f(\mathbf{x}_n | \mathbf{z}_{1:n-1})$. The look-ahead proposal $g(\mathbf{x}_n)$ is not constructed from a previous GMPP—which is not available at the first sensor—but from $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_{1:n-1})$ (cf. Section IV-B). Finally, the GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:1}, \mathbf{z}_{1:n-1})$ to be transmitted to sensor $k=2$ is obtained by sampling from $g(\mathbf{x}_n)$, calculating importance weights, and performing clustering.

B. Gaussian sum filter for proposal design

The look-ahead proposal pdf $g(\mathbf{x}_n)$ can be calculated by two update steps of a *Gaussian sum filter* (GSF) [7], [13]. The GSF input is the received GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ with, possibly, an increased number $M' \geq M$ of GM components (the latter aspect is discussed further below). This GMPP will be denoted by $g_0(\mathbf{x}_n) = \sum_{m=1}^{M'} \alpha_m^{(0)} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_m^{(0)}, \mathbf{C}_m^{(0)})$. At the GSF output, $g(\mathbf{x}_n)$ is obtained as another M' -component GM. This output GM approximates the next PP $f(\mathbf{x}_n | \mathbf{z}_n^{1:k+1}, \mathbf{z}_{1:n-1})$.

The GSF is a bank of M' extended Kalman filters (EKFs); each EKF update step maps the parameters of one input GM component to those of a corresponding output GM component. For simplicity, we assume additive measurement noise, i.e., $\mathbf{h}_n^k(\mathbf{x}_n, \mathbf{w}_n^k) = \tilde{\mathbf{h}}_n^k(\mathbf{x}_n) + \mathbf{w}_n^k$ (see [14] for a treatment of the general case). In what follows, $\mathbf{H}_{n,m}^k \triangleq d\tilde{\mathbf{h}}_n^k(\mathbf{x}_n)/d\mathbf{x}_n|_{\mathbf{x}_n=\boldsymbol{\mu}_m^{(0)}}$ and $\mathbf{C}_{\mathbf{w}_n^k}$ denotes the covariance matrix of \mathbf{w}_n^k .

Step 1: The input GM $g_0(\mathbf{x}_n)$ is updated to incorporate the local measurement \mathbf{z}_n^k , yielding a new M' -component GM

$$g_1(\mathbf{x}_n) = \sum_{m=1}^{M'} \alpha_m^{(1)} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_m^{(1)}, \mathbf{C}_m^{(1)}) \approx \frac{f(\mathbf{z}_n^k | \mathbf{x}_n) g_0(\mathbf{x}_n)}{\int f(\mathbf{z}_n^k | \mathbf{x}_n) g_0(\mathbf{x}_n) d\mathbf{x}_n}. \quad (8)$$

The approximation (8) is achieved by calculating the output parameters $\{\alpha_m^{(1)}, \boldsymbol{\mu}_m^{(1)}, \mathbf{C}_m^{(1)}\}_{m=1}^{M'}$ from the input parameters $\{\alpha_m^{(0)}, \boldsymbol{\mu}_m^{(0)}, \mathbf{C}_m^{(0)}\}_{m=1}^{M'}$ via the EKF update recursions [7], [13]

$$\begin{aligned} \boldsymbol{\mu}_m^{(1)} &= \boldsymbol{\mu}_m^{(0)} + \mathbf{K}_{n,m}^k [\mathbf{z}_n^k - \tilde{\mathbf{h}}_n^k(\boldsymbol{\mu}_m^{(0)})] \\ \mathbf{C}_m^{(1)} &= \mathbf{C}_m^{(0)} - \mathbf{K}_{n,m}^k \mathbf{H}_{n,m}^k \mathbf{C}_m^{(0)} \\ \mathbf{K}_{n,m}^k &= \mathbf{C}_m^{(0)} \mathbf{H}_{n,m}^{k\top} (\mathbf{H}_{n,m}^k \mathbf{C}_m^{(0)} \mathbf{H}_{n,m}^{k\top} + \mathbf{C}_{\mathbf{w}_n^k})^{-1} \end{aligned}$$

and via the recursion $\alpha_m^{(1)} = \alpha_m^{(0)} \beta_{n,m}^k / \sum_{m'=1}^{M'} \alpha_{m'}^{(0)} \beta_{n,m'}^k$, where $\beta_{n,m}^k \triangleq \mathcal{N}(\mathbf{z}_n^k; \tilde{\mathbf{h}}_n^k(\boldsymbol{\mu}_m^{(0)}), \mathbf{H}_{n,m}^k \mathbf{C}_m^{(0)} \mathbf{H}_{n,m}^{k\top} + \mathbf{C}_{\mathbf{w}_n^k})$.

Step 2: The GM $g_1(\mathbf{x}_n)$ just obtained is updated again to incorporate the back-transmitted measurement \mathbf{z}_n^{k+1} , which yields another M' -component GM

$$g_2(\mathbf{x}_n) = \sum_{m=1}^{M'} \alpha_m^{(2)} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_m^{(2)}, \mathbf{C}_m^{(2)}) \approx \frac{f(\mathbf{z}_n^{k+1} | \mathbf{x}_n) f(\mathbf{z}_n^k | \mathbf{x}_n) g_1(\mathbf{x}_n)}{\int f(\mathbf{z}_n^{k+1} | \mathbf{x}_n) f(\mathbf{z}_n^k | \mathbf{x}_n) g_1(\mathbf{x}_n) d\mathbf{x}_n}. \quad (9)$$

The approximation (9) is achieved by calculating the output parameters $\{\alpha_m^{(2)}, \boldsymbol{\mu}_m^{(2)}, \mathbf{C}_m^{(2)}\}_{m=1}^{M'}$ from the input parameters $\{\alpha_m^{(1)}, \boldsymbol{\mu}_m^{(1)}, \mathbf{C}_m^{(1)}\}_{m=1}^{M'}$ via the recursions in Step 1 with \mathbf{z}_n^k replaced by \mathbf{z}_n^{k+1} and with obvious further substitutions.

The GM $g_2(\mathbf{x}_n)$ resulting from Step 2 is used as the look-ahead proposal pdf $g(\mathbf{x}_n)$. According to (9), it approximates the next PP $f(\mathbf{x}_n | \mathbf{z}_n^{1:k+1}, \mathbf{z}_{1:n-1})$, as desired.

If back-transmission of \mathbf{z}_n^{k+1} is not an option, the result $g_1(\mathbf{x}_n)$ of Step 1 can be used as a proposal pdf that still incorporates the current measurement \mathbf{z}_n^k . According to (8), $g_1(\mathbf{x}_n)$ approximates the current PP $f(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$. Hence, sampling from $g_1(\mathbf{x}_n)$ tends to produce particles in regions where the current PP is concentrated, and thus improves on the conventional proposal pdf $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ (i.e., $g_0(\mathbf{x}_n)$). However, as will be shown in Section VI, the performance improvement is not as large as with $g_2(\mathbf{x}_n)$.

To obtain good proposal pdf's $g_2(\mathbf{x}_n)$ or $g_1(\mathbf{x}_n)$, it may be beneficial to increase the number M of GM components in the GSF input $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$. Indeed, this “decreases” the covariance matrices $\mathbf{C}_m^{(0)}$ of $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$, which improves the performance of the GSF [7] and yields also a better approximation of highly non-Gaussian PPs. Note, however, that this does not increase the communication cost since we still transmit only M GM components.

To increase the number of GM components, we draw $M' > M$ samples from $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ and use *kernel density estimation* [15] to construct a new GM with M' components. Thus, for each sample, we obtain one new GM component. The mean of this component is given by the value of the sample, and the covariance is a fixed, suitably chosen covariance matrix with a small trace [15].

We finally note that the EKF can be replaced with the sigma-point Kalman filter, as suggested in [16].

V. APPLICATION: TARGET TRACKING

As an example application, we consider target tracking using acoustic amplitude sensors.

State-space model. The state \mathbf{x}_n now represents the 2D position of a target in a plane, which evolves with time according to (cf. (1)) $\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{u}_n$, $n = 1, 2, \dots$. Here, \mathbf{u}_n is white and uniform over a circular region with radius Tv_{max} centered at $\mathbf{0}$ (T is the duration of one sampling interval and v_{max} is the maximum target speed). It follows that $f(\mathbf{x}_n | \mathbf{x}_{n-1})$ is uniform over a circular region with radius Tv_{max} centered at \mathbf{x}_{n-1} . The target emits a sound of constant amplitude A , which is sensed by acoustic amplitude sensors [1]. Let $\boldsymbol{\xi}_n^k$ denote the position of sensor k at time n . The (scalar) measurement z_n^k of sensor k is given by (cf. (2))

$$z_n^k = \frac{A}{r_n^k} + w_n^k, \quad (10)$$

where $r_n^k \triangleq \|\mathbf{x}_n - \boldsymbol{\xi}_n^k\|$ and $w_n^k \sim \mathcal{N}(0, \sigma_w^2)$ [9]. Thus, $f(z_n^k | \mathbf{x}_n)$ is Gaussian with mean A/r_n^k and variance σ_w^2 .

Distributed sensor scheduling. At each time n , the PPs are propagated through a *local sensor chain* (LSC) consisting of

K_n sensors (cf. Fig. 1). The first sensor of each LSC is also the last sensor of the previous LSC. The LSCs are formed by a distributed scheduling algorithm, based on the principle that the currently active sensor k selects and queries a candidate sensor to be included as the next sensor in the LSC. More specifically, sensor k performs the following steps.

- Receive a GMPP from sensor $k-1$ and perform the PP update described in Section III-A.
- Calculate the local minimum mean-square error (MMSE) estimate of the target position, $\hat{\mathbf{x}}_n^k \triangleq E\{\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1}\} = \int \mathbf{x}_n f(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1}) d\mathbf{x}_n$ (this can be done very easily using the particle representation of $f(\mathbf{x}_n | \mathbf{z}_n^{1:k}, \mathbf{z}_{1:n-1})$).
- Query the neighboring sensor that is closest to $\hat{\mathbf{x}}_n^k$ (we assume that sensor k knows the locations of all its neighbors).
- If the queried sensor accepts to be scheduled, transmit the updated GMPP to that sensor. The sensor then becomes the new active sensor of the LSC.
- If the queried sensor refuses to be scheduled (because it is not operating or it has already been used in the current LSC or it is not sufficiently close to the target as discussed below), query the second-closest neighbor (and so on).

If the local target position estimate $\hat{\mathbf{x}}_n^k$ is inaccurate, the sensor closest to $\hat{\mathbf{x}}_n^k$ may be far from the true target position \mathbf{x}_n and, thus, may have a small local signal-to-noise ratio (SNR), $(z_n^k)^2 / \sigma_w^2$. If that sensor is scheduled as the next sensor of the LSC, the next local estimate $\hat{\mathbf{x}}_n^{k+1}$ will again be inaccurate, etc. This will lead to a poorly chosen LSC and, in turn, a poorly estimated total posterior $f(\mathbf{x}_n | \mathbf{z}_{1:n})$ at the end of the LSC. To mitigate this effect, we propose a *target proximity detection* (TPD) step: each sensor compares its local SNR $(z_n^k)^2 / \sigma_w^2$ to a suitably chosen threshold γ , and, when queried, accepts to be scheduled only if $(z_n^k)^2 / \sigma_w^2 > \gamma$.

VI. SIMULATION RESULTS

We simulated a target tracking experiment with a target emitting a sound of amplitude $A = 100$ and moving with maximum speed $v_{\text{max}} = 40 \text{ m}/T$. The WSN consisted of 2500 acoustic amplitude sensors that were deployed on a grid with randomly jittered grid point positions [9] within a region of size $2000 \text{ m} \times 2000 \text{ m}$. Each sensor communicated with other sensors within a range of 80m. The LSC length was $K_n = 4$ for all n . Furthermore, we used the following parameters unless noted otherwise: $S = 500$, $M = 4$, $M' = 1000$, $\sigma_w = 0.1$.

We compare the following three versions of the proposed distributed PF with GM transmissions. The version denoted *GM-0* uses the proposal pdf $g_0(\mathbf{x}_n)$, i.e., the received GMPP $f_{\text{GM}}(\mathbf{x}_n | \mathbf{z}_n^{1:k-1}, \mathbf{z}_{1:n-1})$ as suggested in [4]. *GM-1* uses the proposal pdf $g_1(\mathbf{x}_n)$ incorporating the local measurement \mathbf{z}_n^k , and *GM-2* uses the look-ahead proposal pdf $g_2(\mathbf{x}_n)$ (see Section IV-B). We also consider two conventional PFs that do not use GM transmissions: *DPF* is a distributed PF that transmits the complete particle representation of the PPs; *CPF* is a clairvoyant centralized PF in which a fusion center processes the measurements of the 4 ($= K_n$) sensors closest to the *true*

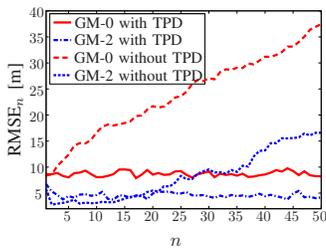


Figure 3. $RMSE_n$ versus time.

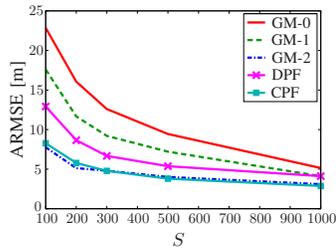


Figure 4. ARMSE versus number of particles.

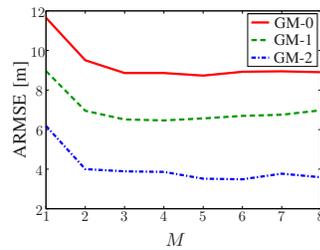


Figure 5. ARMSE versus number of transmitted GM components.

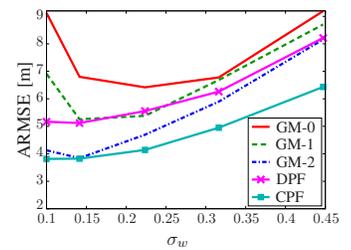


Figure 6. ARMSE versus standard deviation of measurement noise.

target position by means of the auxiliary PF algorithm [17].

As a performance measure for GM-0/1/2 and DPF, we use the n -dependent root mean-square error (denoted $RMSE_n$) of the MMSE state estimate at the last sensor $k = K_n = 4$ of the LSC at time n (calculated from the local particle representation of $f(\mathbf{x}_n | \mathbf{z}_n^{1:4}, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n | \mathbf{z}_{1:n})$), with averaging performed over 1500 simulation runs. For CPF, the calculation of the $RMSE_n$ was based on the MMSE state estimate obtained at the fusion center. We also computed the *average RMSE* (ARMSE) by averaging $(RMSE_n)^2$ over all 50 simulated time instants and taking the square root of the result.

Fig. 3 shows the temporal (n) evolution of $RMSE_n$ for GM-0 and GM-2 using sensor scheduling with and without TPD (see Section V). It is seen that GM-2 (i.e., the look-ahead scheme) performs significantly better than GM-0. Furthermore, in both cases, TPD prevents a systematic increase of the $RMSE_n$ with n . Therefore, TPD was used in the simulations reported in Figs. 4–6 (except for CPF, where no sensor scheduling is done).

In Fig. 4, we show the dependence of the ARMSE on the number S of particles for all five estimation methods. As expected, all methods perform better when S is larger. GM-2 again yields a significant performance improvement over GM-0. In fact, the results of GM-2 are as good as those of the clairvoyant CPF or even slightly better. GM-1, too, performs better than GM-0 but still worse than GM-2, DPF, and CPF.

The dependence of the ARMSE of GM-0/1/2 on the number M of transmitted GM components is depicted in Fig. 5. We conclude that no significant performance gains are obtained by increasing M beyond 3.

Finally, Fig. 6 displays the ARMSE of all methods versus the measurement noise level. It is seen that the performance gains of GM-2 are largest in the low-noise regime: since for low noise the likelihoods are very peaky, it is important to draw samples from proposals that take the measurements into account.

VII. CONCLUSION

We presented a scheme for distributed, time-space-sequential, Bayesian state estimation in wireless sensor networks. Posterior densities are transmitted from one sensor to the next and updated at each sensor using the local measurements. The scheme requires communication only between neighboring sensors. Following [4], particle filter techniques are used for an efficient implementation of the update and prediction steps at the sensors, and low transmission rates are achieved

by transmitting Gaussian mixture (GM) representations rather than particles. The GM representations are calculated by k-means clustering. We introduced a “look-ahead” design of proposal densities that yields improved GM approximations. The performance of the proposed estimation scheme, combined with a decentralized sensor scheduling algorithm, was assessed for a target tracking application. Our results demonstrated the superiority of the proposed look-ahead design.

REFERENCES

- [1] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Amsterdam, The Netherlands: Morgan Kaufmann, 2004.
- [2] A. Doucet, N. De Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer, 2001.
- [3] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, “Particle filtering,” *IEEE Signal Process. Mag.*, vol. 20, pp. 19–38, Sep. 2003.
- [4] X. Sheng, Y. H. Hu, and P. Ramanathan, “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network,” in *Proc. IPSN-05*, (Los Angeles, CA), pp. 181–188, Apr. 2005.
- [5] O. Hlinka and F. Hlawatsch, “Time-space-sequential algorithms for distributed Bayesian state estimation in serial sensor networks,” in *Proc. IEEE ICASSP-09*, (Taipei, Taiwan), pp. 2057–2060, Apr. 2009.
- [6] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.
- [7] D. Alspach and H. Sorenson, “Nonlinear Bayesian estimation using Gaussian sum approximations,” *IEEE Trans. Autom. Control*, vol. 17, pp. 439–448, Aug. 1972.
- [8] T. Zhao and A. Nehorai, “Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 55, pp. 1511–1524, Apr. 2007.
- [9] J. Liu, J. Reich, and F. Zhao, “Collaborative in-network processing for target tracking,” *EURASIP J. Appl. Sig. Process.*, vol. 2003, pp. 378–391, Jan. 2003.
- [10] M. Coates, “Distributed particle filters for sensor networks,” in *Proc. IPSN-04*, (Berkeley, CA), pp. 99–107, Apr. 2004.
- [11] M. Rosencrantz, G. Gordon, and S. Thrun, “Decentralized sensor fusion with distributed particle filters,” in *Proc. UAI-03*, (Acapulco, Mexico), Aug. 2003.
- [12] B. Ristic and S. Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House, 2004.
- [13] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.
- [14] G. Welch and G. Bishop, “An introduction to the Kalman filter,” tech. rep., TR 95-041, Univ. of North Carolina at Chapel Hill, NC, Jul. 2006.
- [15] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL: Chapman & Hall/CRC, 1986.
- [16] R. van der Merwe and E. Wan, “Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models,” in *Proc. ICASSP-03*, (Hong Kong, China), pp. VI 701–704, Apr. 2003.
- [17] M. K. Pitt and N. Shephard, “Filtering via simulation: Auxiliary particle filters,” *J. Am. Stat. Assoc.*, vol. 94, pp. 590–599, Jun. 1999.