Distributed Sequential Estimation in Asynchronous Wireless Sensor Networks

Ondrej Hlinka, Franz Hlawatsch, *Fellow, IEEE*, and Petar M. Djurić, *Fellow, IEEE*

***Abstract*— We propose a distributed sequential estimation scheme for wireless sensor networks with asynchronous measurements. Our scheme combines the prediction and update steps of a Bayesian filter (for time alignment and recursive state estimation) with a fusion rule (for intersensor fusion using local communication). We also propose a reduced-complexity implementation using particle filtering and Gaussian mixture approximations, and an estimator of the delays resulting from processing and communication. Simulations for a target tracking problem demonstrate the good performance of our scheme.**

***Index Terms*— Wireless sensor network, asynchronous measurements, data fusion, distributed state estimation, distributed particle filter, target tracking.**

I. Introduction

Distributed (decentralized) sequential estimation in wireless sensor networks has many applications [1]–[3]. In practice, synchronizing all the sensors is often difficult, and thus the measurements are taken asynchronously. Most estimation schemes for asynchronous measurements are centralized, i.e., they rely on a fusion center where the measurements are collected and processed [4]–[8]. In the distributed method proposed in [9], each sensor broadcasts its local measurements to all the other sensors in the network. To the best of our knowledge, a truly distributed estimation scheme for asynchronous measurements in which each sensor communicates only with neighbor sensors has not been proposed so far.

Here, we present a distributed sequential Bayesian estimation scheme for asynchronous measurements in which local posterior probability density functions (pdfs) are exchanged between neighbor sensors. Our scheme combines the prediction and update steps of a Bayesian filter with a fusion rule. We also propose a reduced-complexity implementation using particle filtering and Gaussian mixture approximations, and an estimator of the processing-and-communication delays.

This paper is organized as follows. The system model is described in Section II. In Section III, a distributed scheme for asynchronous sequential estimation is proposed. In Section IV, an estimator of the processing-and-communication delays is presented. A particle implementation of our scheme is described in Section V. Finally, simulation results for a target tracking problem are reported in Section VI.

O. Hlinka is with RobArt GmbH, 4020 Linz, Austria (e-mail: ondrej.hlinka@robart.cc). F. Hlawatsch is with the Institute of Telecommunications, Vienna University of Technology, A-1040 Vienna, Austria (e-mail:franz.hlawatsch@nt.tuwien.ac.at). P. M. Djurić is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA (e-mail: petar.djuric@stonybrook.edu). This work was supported by the FWF under Awards S10603 and P27370 and by the NSF under Awards CCF-1320626, ECCS-1346854, and CNS-1354614.

II. System Model

We consider a state vector $\mathbf{x}(t) = (x_1(t) \cdots x_M(t))^\top \in \mathbb{R}^M$ that evolves with continuous time $t \in \mathbb{R}_+$ according to (cf. [6], [9])

$$\mathbf{x}(t) = \mathbf{a}_{t',t}(\mathbf{x}(t'), \mathbf{u}_{t',t}) \,, \tag{1}$$

for any $t, t' \in \mathbb{R}_+$ with $t > t'$. Here, $\mathbf{a}_{t',t}(\cdot\,,\cdot)$ is a generally nonlinear function and $\mathbf{u}_{t',t}$ is driving noise with a known pdf $f(\mathbf{u}_{t',t})$. The state $\mathbf{x}(t)$ is sensed via asynchronous measurements by a wireless sensor network consisting of K sensors. Sensor $k \in \{1, \ldots, K\}$ acquires its ith measurement $\mathbf{z}_{k,i} \in \mathbb{R}^{N_{k,i}}$, $i \in \mathbb{N}$ at time $t = t_{k,i}$ according to

$$\mathbf{z}_{k,i} = \mathbf{h}_{k,i}(\mathbf{x}(t_{k,i}), \mathbf{v}_{k,i}) \,, \tag{2}$$

where $\mathbf{h}_{k,i}(\cdot\,,\cdot)$ is a generally nonlinear function and $\mathbf{v}_{k,i}$ is measurement noise with a known pdf $f(\mathbf{v}_{k,i})$. We assume that $\mathbf{v}_{k,i}$ and $\mathbf{v}_{k',i'}$ are independent unless $(k,i) = (k',i')$; the initial state $\mathbf{x}(0)$ and the noises $\mathbf{u}_{t',t}$ and $\mathbf{v}_{k,i}$ are independent; and sensor k knows $\mathbf{h}_{k,i}(\cdot\,,\cdot)$ and $\mathbf{a}_{t',t}(\cdot\,,\cdot)$ for all t', t with $0 \leq t' < t$. We denote by $\mathcal{N}_k$ the set of "neighbor" sensors with which sensor k is able to communicate.

The state-space model (1), (2) and our statistical assumptions determine the *state-transition pdf* $f(\mathbf{x}(t) \,|\, \mathbf{x}(t'))$ and the *local likelihood functions* $f(\mathbf{z}_{k,i} \,|\, \mathbf{x}(t_{k,i}))$.

III. Distributed Asynchronous Estimation

Let us denote by $f_k\big(\mathbf{x}(t) \,\big|\, \{\mathbf{z}\}_k^t\big)$ the "local" posterior pdf (abbreviated as LP) at sensor k. Here, $\{\mathbf{z}\}_k^t$ denotes all the measurements of sensor k and other sensors that were incorporated into the LP until time instant t. From the LP $f_k\big(\mathbf{x}(t) \,\big|\, \{\mathbf{z}\}_k^t\big)$, sensor k is able to calculate a minimum mean-square error (MMSE) estimate of the state $\mathbf{x}(t)$ [10],

$$\hat{\mathbf{x}}(t) \triangleq \mathrm{E}\big\{\mathbf{x}(t) \,\big|\, \{\mathbf{z}\}_k^t\big\} = \int \mathbf{x}(t)\, f_k\big(\mathbf{x}(t) \,\big|\, \{\mathbf{z}\}_k^t\big)\, d\mathbf{x}(t) \,. \tag{3}$$

The proposed distributed sequential estimation scheme consists of two main functionalities: (i) incorporating a local measurement into the LP and (ii) fusing the LP with the LP of a neighbor sensor. These functionalities are described next.

A. Incorporating a Local Measurement

Let us assume that the LP $f_k\big(\mathbf{x}(t)\big|\{\mathbf{z}\}_k^t\big)$ is available at sensor k at time t, and let $\mathbf{z}_{k,i}$ denote the first measurement that sensor k acquires after time t. Sensor k then incorporates $\mathbf{z}_{k,i}$ into $f_k\big(\mathbf{x}(t)\big|\{\mathbf{z}\}_k^t\big)$ by performing the prediction and update steps of a Bayesian filter [11]. First, sensor k predicts its LP from time t to the time when it acquires $\mathbf{z}_{k,i}$, $t_{k,i} > t$. The resulting predictive pdf of $\mathbf{x}(t_{k,i})$ is given by

$$f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^t\big) = \int f(\mathbf{x}(t_{k,i})\,|\,\mathbf{x}(t))\, f_k\big(\mathbf{x}(t)\big|\{\mathbf{z}\}_k^t\big)\, d\mathbf{x}(t)\,. \tag{4}$$

Next, sensor k calculates the LP at time $t_{k,i}$ by incorporating its measurement $\mathbf{z}_{k,i}$ into the predictive pdf according to

$$f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big) \propto f(\mathbf{z}_{k,i}\,|\,\mathbf{x}(t_{k,i}))\, f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^t\big)\,. \tag{5}$$

Note that $\{\mathbf{z}\}_k^{t_{k,i}} = \{\mathbf{z}\}_k^t \cup \{\mathbf{z}_{k,i}\}$. Sensor k is now able to calculate from $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ the MMSE state estimate at time $t_{k,i}$, $\hat{\mathbf{x}}(t_{k,i})$ (cf. (3)). Finally, sensor k broadcasts its new LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ to its neighbors $l \in \mathcal{N}_k$.

B. LP Fusion

According to Section III-A, neighbor sensor $l \in \mathcal{N}_k$ receives $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ from sensor k at some time $t'_{k,l,i} = t_{k,i} + \tau_{k,l}$. Here, $\tau_{k,l} > 0$ is the time used for processing at sensor k and transmission from sensor k to sensor l. We assume that (i) the delay $\tau_{k,l}$ is small in the sense that $\tau_{k,l} < t_{l',i'} - t_{k,i}$ for any $t_{l',i'} > t_{k,i}$ with $l' \in \mathcal{N}_l \cup \{l\}$ (equivalently, $t_{l',i'} > t_{k,i} + \tau_{k,l}$, i.e., a neighbor l' of sensor k may acquire its next measurement only after receiving the LP from sensor k), and (ii) sensor l knows $\tau_{k,l}$ and can thus infer from $t'_{k,l,i}$ the time $t_{k,i}$ underlying $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$. The estimation of $\tau_{k,l}$ will be addressed in Section IV.

Let $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$ denote the LP of sensor l that is available at time $t'_{k,l,i}$, and which was calculated by sensor l at some past time t'. Our above assumption (i) implies that $t' < t_{k,i}$ (i.e., the received LP is "newer" than the current LP of sensor l). The next step for sensor l is to fuse its LP $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$ with the LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ received from sensor k. Because the two LPs correspond to different times $t' < t_{k,i}$ and $t_{k,i}$, they first need to be time-aligned. Sensor l thus performs a prediction step to calculate the predictive pdf of $\mathbf{x}(t_{k,i})$ from its LP $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$:

$$\begin{aligned} &f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big) \\ &\quad = \int f(\mathbf{x}(t_{k,i})\,|\,\mathbf{x}(t'))\, f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)\, d\mathbf{x}(t')\,. \end{aligned} \tag{6}$$

Sensor l then fuses its predictive pdf $f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ with the received LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$, thereby obtaining a new LP $f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t_{k,i}}\big)$ with $\{\mathbf{z}\}_l^{t_{k,i}} = \{\mathbf{z}\}_l^{t'} \cup \{\mathbf{z}\}_k^{t_{k,i}}$. (This new LP is not transmitted to the neighbors of sensor l.)

The measurement sets of the two LPs may overlap, i.e., $\{\mathbf{z}\}_l^{t'} \cap \{\mathbf{z}\}_k^{t_{k,i}} \neq \emptyset$. Indeed, since the communication graph generally contains loops and the communication paths are not controlled, it is not possible to remove double measurements before fusing two LPs. A "conservative" fusion of the two LPs $f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ and $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ without explicit double-counting of common past measurements is enabled by the geometric mean density (GMD) fusion rule [12]

$$\begin{aligned} &f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t_{k,i}}\big) \\ &\quad \propto \big[f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)\big]^{\omega} \big[f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)\big]^{1-\omega}. \end{aligned} \tag{7}$$

Here, the choice of $\omega \in [0,1]$ is discussed in [12], [13]. Via (7), sensor l obtains a new LP that is based on the union of the measurement sets of sensors l and k. By using this scheme, the information gradually diffuses through the network. We note that in order to prevent double-counting of common past measurements, the GMD fusion rule underweights the new information, which can lead to some performance degradation. Alternative fusion rules can be found in [14], [15].

If assumption (i) is not met, it may happen that $t' > t_{k,i}$, i.e., the LP $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$ may be "newer" than the received LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$). Here, instead of performing a prediction step on its own LP, sensor l predicts the received LP (cf. (6)); the obtained predictive pdf and the LP of sensor l are then fused using the GMD fusion rule (cf. (7)). However, to distinguish between the two cases $t' < t_{k,i}$ and $t' > t_{k,i}$, sensor l would need to know $t_{k,i}$. Imparting $t_{k,i}$ to sensor l by transmitting a local clock value (time stamp) from sensor k to sensor l is possible only if the local clock skews are equal and the communication delay is negligible or known.

For good performance of our scheme, long measurement-free periods at individual sensors should be avoided. Otherwise, a part of the network does not receive any posterior updates for a long time, which may lead to long prediction horizons and, in turn, poor estimation performance.

IV. Delay Estimation

Next, we develop an estimator of the processing-and-communication delay $\tau_{k,l}$. Estimation of $\tau_{k,l}$ is performed at sensor l after the time instant $t'_{k,l,i}$ at which sensor l receives the LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ from sensor k. As in Section III-B, the LP available at sensor l at time $t'_{k,l,i}$ is $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$, which was calculated by sensor l at time t'. We again use assumption (i) stated in Section III-B, which ensures that $t' < t_{k,i}$. We recall that $t'_{k,l,i} = t_{k,i} + \tau_{k,l}$, with $\tau_{k,l} > 0$.

We model $\tau_{k,l}$ as a random variable with a prior pdf $f(\tau_{k,l})$ (e.g., an exponential pdf). To construct a likelihood function for $\tau_{k,l}$, we rewrite (1) for $t = t_{k,i} = t'_{k,l,i} - \tau_{k,l}$, i.e.,

$$\mathbf{x}(t_{k,i}) \,=\, \mathbf{a}_{t',t'_{k,l,i}-\tau_{k,l}}\big(\mathbf{x}(t'), \mathbf{u}_{t',t'_{k,l,i}-\tau_{k,l}}\big)\,. \tag{8}$$

Because $\mathbf{x}(t_{k,i})$ and $\mathbf{x}(t')$ are unknown, we express them in terms of their MMSE estimates $\hat{\mathbf{x}}(t_{k,i})$ and $\hat{\mathbf{x}}(t')$, i.e., we write $\mathbf{x}(t_{k,i}) = \hat{\mathbf{x}}(t_{k,i}) + \mathbf{e}_{t_{k,i}}$ and $\mathbf{x}(t') = \hat{\mathbf{x}}(t') + \mathbf{e}_{t'}$, where $\mathbf{e}_{t_{k,i}}$ and $\mathbf{e}_{t'}$ are estimation errors. Equation (8) then becomes

$$\hat{\mathbf{x}}(t_{k,i}) \,=\, \mathbf{a}_{t',t'_{k,l,i}-\tau_{k,l}}\big(\hat{\mathbf{x}}(t') + \mathbf{e}_{t'}, \mathbf{u}_{t',t'_{k,l,i}-\tau_{k,l}}\big) - \mathbf{e}_{t_{k,i}}\,. \tag{9}$$

Sensor l calculates $\hat{\mathbf{x}}(t_{k,i})$ from $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ and $\hat{\mathbf{x}}(t')$ from $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$ (cf. (3)). Since the dependencies between $\mathbf{u}_{t',t'_{k,l,i}-\tau_{k,l}}$, $\mathbf{e}_{t_{k,i}}$, and $\mathbf{e}_{t'}$ are unknown, we model these random vectors as independent, with $\mathbf{e}_{t_{k,i}} \sim \mathcal{N}(\mathbf{0},$

$\mathbf{C}_{\mathbf{e}_{t_{k,i}}}$) and $\mathbf{e}_{t'} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{\mathbf{e}_{t'}})$. In theory (cf. [10, Eq. (14.3)]), $\mathbf{C}_{\mathbf{e}_{t_{k,i}}} = \mathrm{E}_{\{\mathbf{z}\}_k^{t_{k,i}}}\big\{\mathrm{cov}\big\{\mathbf{x}(t_{k,i})|\{\mathbf{z}\}_k^{t_{k,i}}\big\}\big\}$ and $\mathbf{C}_{\mathbf{e}_{t'}} = \mathrm{E}_{\{\mathbf{z}\}_l^{t'}}\big\{\mathrm{cov}\big\{\mathbf{x}(t')|\{\mathbf{z}\}_l^{t'}\big\}\big\}$, where $\mathrm{cov}\big\{\mathbf{x}(t_{k,i})|\{\mathbf{z}\}_k^{t_{k,i}}\big\}$ denotes the covariance matrix of the LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ and $\mathrm{cov}\big\{\mathbf{x}(t')|\{\mathbf{z}\}_l^{t'}\big\}$ that of the LP $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$. However, since computing the above expectations is infeasible, we set $\mathbf{C}_{\mathbf{e}_{t_{k,i}}}$ and $\mathbf{C}_{\mathbf{e}_{t'}}$ equal to the covariance matrices of the locally available (approximate) LPs, i.e., $\mathbf{C}_{\mathbf{e}_{t_{k,i}}} = \mathrm{cov}\big\{\mathbf{x}(t_{k,i})|\{\mathbf{z}\}_k^{t_{k,i}}\big\}$ and $\mathbf{C}_{\mathbf{e}_{t'}} = \mathrm{cov}\big\{\mathbf{x}(t')|\{\mathbf{z}\}_l^{t'}\big\}$. We can then interpret (9) as a measurement model, where $\tau_{k,l}$ is the parameter to be estimated and $\hat{\mathbf{x}}(t_{k,i})$ is interpreted as an observed measurement. This measurement model, together with the statistical properties of $\mathbf{u}_{t',t'_{k,l,i}-\tau_{k,l}}$, $\mathbf{e}_{t_{k,i}}$, and $\mathbf{e}_{t'}$, determines the likelihood function of $\tau_{k,l}$, $f(\hat{\mathbf{x}}(t_{k,i})|\tau_{k,l})$. (This is analogous to obtaining the local likelihood function $f(\mathbf{z}_{k,i}\,|\,\mathbf{x}(t_{k,i}))$ from (2).)

Using this likelihood function and the prior pdf $f(\tau_{k,l})$, sensor l calculates the posterior pdf of $\tau_{k,l}$ as $f(\tau_{k,l}|\hat{\mathbf{x}}(t_{k,i})) \propto f(\hat{\mathbf{x}}(t_{k,i})|\tau_{k,l})\, f(\tau_{k,l})$. From the posterior pdf, sensor l calculates an MMSE estimate $\hat{\tau}_{k,l}$ (cf. (3)) and, in turn, an estimate of $t_{k,i}$ as $\hat{t}_{k,i} = t'_{k,l,i} - \hat{\tau}_{k,l}$. This latter estimate is used by sensor l for LP fusion (see Section III-B). A practical implementation can be obtained via importance sampling [16]. Samples $\big\{\tau_{k,l}^{(j)}\big\}_{j=1}^{J}$ are drawn from the prior pdf $f(\tau_{k,l})$, and importance weights are calculated as $w^{(j)} = f\big(\hat{\mathbf{x}}(t_{k,i})\big|\tau_{k,l}^{(j)}\big)/c$, $j = 1, \ldots, J$, with $c = \sum_{j=1}^{J} f\big(\hat{\mathbf{x}}(t_{k,i})\big|\tau_{k,l}^{(j)}\big)$. The MMSE estimate of $\tau_{k,l}$ is then approximated as $\hat{\tau}_{k,l} \approx \sum_{j=1}^{J} w^{(j)}\tau_{k,l}^{(j)}$.

V. Particle Filtering Implementation

Next, we present a reduced-complexity, approximate implementation of our distributed sequential estimation scheme using particle filtering [11], [16]–[18]. In a particle filter, the posterior pdf $f\big(\mathbf{x}(t)\big|\{\mathbf{z}\}^{t}\big)$ is represented by samples (particles) $\big\{\mathbf{x}(t)^{(j)}\big\}_{j=1}^{J}$ and corresponding weights $\big\{w(t)^{(j)}\big\}_{j=1}^{J}$. Furthermore, to reduce the communication cost, we use a Gaussian mixture (GM) approximation of the posterior pdfs transmitted between neighbor sensors. In what follows, the proposed algorithm is stated for a generic sensor k that acquires a new measurement and a generic neighbor sensor $l \in \mathcal{N}_k$ that fuses the updated LP of sensor k with its own LP.

Distributed Asynchronous Particle Filter Algorithm

Part 1 – Operations executed by sensor k (cf. Section III-A): At time t, sensor k has available a particle representation $\big\{\big(\mathbf{x}(t)^{(j)}, w(t)^{(j)}\big)\big\}_{j=1}^{J}$ of its LP $f_k\big(\mathbf{x}(t)\big|\{\mathbf{z}\}_k^{t}\big)$. When sensor k obtains (at some time $t_{k,i} > t$) a new measurement $\mathbf{z}_{k,i}$, it performs the following steps:

1.1 New particles $\big\{\mathbf{x}(t_{k,i})^{(j)}\big\}_{j=1}^{J}$ are drawn from $f\big(\mathbf{x}(t_{k,i})\big|\mathbf{x}(t)^{(j)}\big) \equiv f(\mathbf{x}(t_{k,i})|\mathbf{x}(t))\big|_{\mathbf{x}(t)=\mathbf{x}(t)^{(j)}}$. (This step corresponds to the prediction step (4).)

1.2 Weights associated with the new particles $\mathbf{x}(t_{k,i})^{(j)}$ are calculated as $w(t_{k,i})^{(j)} = f\big(\mathbf{z}_{k,i}\big|\mathbf{x}(t_{k,i})^{(j)}\big)/c$, with $c = \sum_{j=1}^{J} f\big(\mathbf{z}_{k,i}\big|\mathbf{x}(t_{k,i})^{(j)}\big)$. (This step corresponds to the update step (5).) We note that $\big\{\big(\mathbf{x}(t_{k,i})^{(j)}, w(t_{k,i})^{(j)}\big)\big\}_{j=1}^{J}$ constitutes a particle representation of the LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$.

1.3 An approximation of the MMSE estimate (cf. (3)) is computed as $\hat{\mathbf{x}}(t_{k,i}) \approx \sum_{j=1}^{J} w(t_{k,i})^{(j)}\mathbf{x}(t_{k,i})^{(j)}$.

1.4 A GM representation $f_k^{\mathrm{GM}}\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ of the LP $f_k\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ is computed from the particle representation $\big\{\big(\mathbf{x}(t_{k,i})^{(j)}, w(t_{k,i})^{(j)}\big)\big\}_{j=1}^{J}$ by means of the weighted EM algorithm [19], and the GM parameters are broadcast to all neighbor sensors $l \in \mathcal{N}_k$.

Part 2 – Operations executed by each sensor $l \in \mathcal{N}_k$ (cf. Section III-B): Sensor l receives the parameters of $f_k^{\mathrm{GM}}\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ from sensor k at time $t'_{k,l,i} = t_{k,i} + \tau_{k,l}$. At that time, it has available a particle representation $\big\{\big(\mathbf{x}(t')^{(j)}, w(t')^{(j)}\big)\big\}_{j=1}^{J}$ of its LP $f_l\big(\mathbf{x}(t')\big|\{\mathbf{z}\}_l^{t'}\big)$, where $t' < t_{k,i}$. Sensor l now performs the following steps:

2.1 An estimate $\hat{t}_{k,i}$ of $t_{k,i}$ is calculated (see Section IV).

2.2 From the LP particle representation $\big\{\big(\mathbf{x}(t')^{(j)}, w(t')^{(j)}\big)\big\}_{j=1}^{J}$, a particle representation $\big\{\big(\mathbf{x}(\hat{t}_{k,i})^{(j)}, w(\hat{t}_{k,i})^{(j)}\big)\big\}_{j=1}^{J}$ of the predictive pdf $f_l\big(\mathbf{x}(\hat{t}_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ is obtained: the particles $\big\{\mathbf{x}(\hat{t}_{k,i})^{(j)}\big\}_{j=1}^{J}$ are drawn from $f\big(\mathbf{x}(\hat{t}_{k,i})\big|\mathbf{x}(t')^{(j)}\big)$ and the weights are still $w(\hat{t}_{k,i})^{(j)} = w(t')^{(j)}$ for all j. (This step corresponds to the prediction step (6).)

2.3 A GM representation $f_l^{\mathrm{GM}}\big(\mathbf{x}(\hat{t}_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ of the predictive pdf $f_l\big(\mathbf{x}(\hat{t}_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ is computed from the particle representation $\big\{\big(\mathbf{x}(\hat{t}_{k,i})^{(j)}, w(\hat{t}_{k,i})^{(j)}\big)\big\}_{j=1}^{J}$ by means of the weighted EM algorithm [19].

2.4 The GM representations $f_l^{\mathrm{GM}}\big(\mathbf{x}(\hat{t}_{k,i})\big|\{\mathbf{z}\}_l^{t'}\big)$ and $f_k^{\mathrm{GM}}\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_k^{t_{k,i}}\big)$ are fused using the GM-based implementation of the GMD fusion rule (7) presented in [20]. This yields a GM representation of the fused LP $f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t_{k,i}}\big)$ in (7).

2.5 A particle representation $\big\{\big(\mathbf{x}(t_{k,i})^{(j)}, w(t_{k,i})^{(j)} \equiv 1/J\big)\big\}_{j=1}^{J}$ of $f_l\big(\mathbf{x}(t_{k,i})\big|\{\mathbf{z}\}_l^{t_{k,i}}\big)$ is obtained by sampling from the GM representation computed in Step 2.4.

Part 1 of this algorithm is executed whenever a sensor acquires a new measurement; Part 2 is executed whenever it receives an LP from a neighbor sensor. Only Step 1.4 requires communication with neighbor sensors; all the other steps are performed locally at sensor k or l. Assuming a GM with L components and a state of dimension M, the number of real values broadcast in Step 1.4 is $L\big(M + M(M+1)/2 + 1\big) - 1$.

VI. Simulation Results

We consider a target tracking application in which the state vector $\mathbf{x}(t) = \big(x(t) \;\; y(t) \;\; \dot{x}(t) \;\; \dot{y}(t)\big)^{\top}$ represents the two-dimensional position and velocity of a target. Following [6] and [9], the evolution of $\mathbf{x}(t)$ is modeled as (cf. (1))

$$\mathbf{x}(t) = \mathbf{A}_{t',t}\mathbf{x}(t') + \mathbf{u}_{t',t}\,, \tag{10}$$

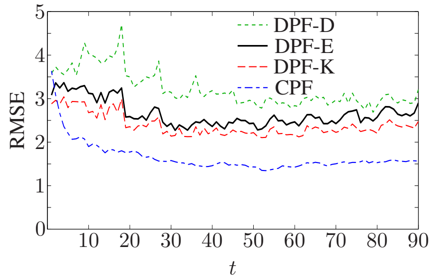

Fig. 1: RMSE versus time.

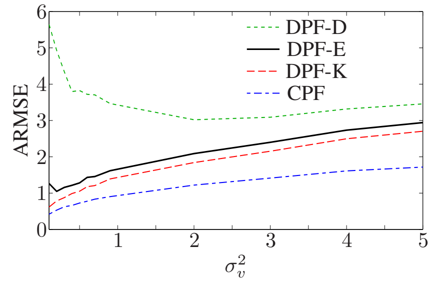

Fig. 2: ARMSE versus the measurement noise variance.

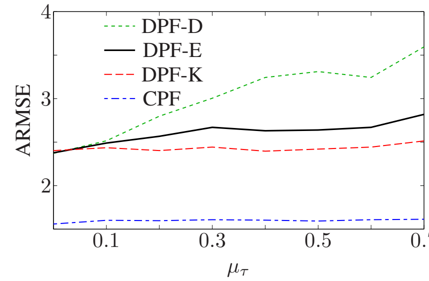

Fig. 3: ARMSE versus the mean processing-and-communication delay.

with $\mathbf{u}_{t',t} \sim \mathcal{N}(\mathbf{0}_4, \mathbf{C}_{t',t})$. Here,

$$\mathbf{A}_{t',t} = \begin{pmatrix} \mathbf{I}_2 & (t-t')\mathbf{I}_2 \\ \mathbf{0}_{2\times 2} & \mathbf{I}_2 \end{pmatrix}, \quad \mathbf{C}_{t',t} = \sigma_u^2 \begin{pmatrix} \Delta_3 & 0 & \Delta_2 & 0 \\ 0 & \Delta_3 & 0 & \Delta_2 \\ \Delta_2 & 0 & \Delta_1 & 0 \\ 0 & \Delta_2 & 0 & \Delta_1 \end{pmatrix},$$

with $\Delta_j = (t-t')^j/j$ and $\sigma_u^2 = 0.001$. The network consists of $K = 9$ sensors deployed on a jittered grid within a square of size 500×500. Each sensor communicates with other sensors within a radius of 300. The measurement $z_{k,i}$ acquired by sensor k at time $t = t_{k,i}$ is given by $z_{k,i} = \big\| \big(x(t_{k,i}) \;\, y(t_{k,i}) \big)^\top - \boldsymbol{\xi}_k \big\| + v_{k,i}$ (cf. (2)), where $v_{k,i} \sim \mathcal{N}(0, \sigma_v^2)$, $\boldsymbol{\xi}_k$ is the position of sensor k and, unless noted otherwise, $\sigma_v^2 = 4$. During each simulation run, sensor k acquires measurements at ten equally spaced time instants $t_{k,i}$ ($i = 1, \ldots, 10$), with $t_{k,i} - t_{k,i-1} = 8$. Different sensors acquire a measurement one after another such that $t_{k,i} - t_{k-1,i} = 1$.

The processing-and-communication delays $\tau_{k,l}$ are randomly generated from an exponential distribution with mean (unless noted otherwise) $\mu_\tau = 0.44$. Rewriting (10) in the form (9) and using the assumptions of Section IV, the likelihood function used for delay estimation is obtained as

$$f(\hat{\mathbf{x}}(t_{k,i}) | \tau_{k,l}) \propto \exp\!\Big(\big[\hat{\mathbf{x}}(t_{k,i}) - \mathbf{A}_{t',t'_{k,l,i}-\tau_{k,l}} \hat{\mathbf{x}}(t') \big]^\top \mathbf{C}^{-1} \cdot \big[\hat{\mathbf{x}}(t_{k,i}) - \mathbf{A}_{t',t'_{k,l,i}-\tau_{k,l}} \hat{\mathbf{x}}(t') \big] \Big),$$

with

$$\mathbf{C} = \mathbf{C}_{t',t'_{k,l,i}-\tau_{k,l}} + \mathbf{A}_{t',t'_{k,l,i}-\tau_{k,l}} \mathbf{C}_{\mathbf{e}_{t'}} \mathbf{A}^\top_{t',t'_{k,l,i}-\tau_{k,l}} + \mathbf{C}_{\mathbf{e}_{t_{k,i}}}.$$

Here, $\mathbf{C}_{\mathbf{e}_{t_{k,i}}}$ and $\mathbf{C}_{\mathbf{e}_{t'}}$ were defined in Section IV; $\mathbf{C}_{\mathbf{e}_{t_{k,i}}}$ is calculated from the GM representation of the LP $f_k\big(\mathbf{x}(t_{k,i}) \big| \{\mathbf{z}\}_k^{t_{k,i}}\big)$ and $\mathbf{C}_{\mathbf{e}_{t'}}$ is calculated from the particle representation of the LP $f_l\big(\mathbf{x}(t') \big| \{\mathbf{z}\}_l^{t'}\big)$.

We simulated three different versions of the proposed distributed asynchronous particle filter: with estimation of the processing-and-communication delays according to Section IV (abbreviated DPF-E), using knowledge of the true delays (DPF-K), and disregarding the delays (DPF-D). In addition, we simulated an idealized centralized asynchronous PF (CPF) that processes each sensor measurement immediately after it is received at a fusion center. This CPF is idealized because there are no processing-and-communication delays. It is equivalent to the "asynchronous sequential filter" proposed in [9] except that the measurements are collected at a fusion center rather than broadcast to all the other sensors. The number of particles used at each sensor (for the DPFs) and at the fusion center (for the CPF) is $J = 5000$. The transmitted posterior pdfs are represented by GMs with four components.

Fig. 1 shows the root-mean-square error (RMSE) of the target position estimates of the various filters versus time t. The RMSE was evaluated at the time instants $t = t_{k,i}$ at which the sensors obtained a measurement, and computed by averaging over 200 simulation runs. The results suggest that the performance of the filters remains stable with progressing time; this was corroborated by further simulations. Fig. 2 shows the time-averaged RMSE (ARMSE) versus the measurement noise variance σ_v^2. We can see from Figs. 1 and 2 that the RMSE of DPF-E is significantly lower than that of DPF-D and close to that of DPF-K; this demonstrates the importance of taking processing-and-communication delays into account and, also, the good performance of our delay estimation scheme. For higher σ_v^2, the performance advantage of DPF-E over DPF-D becomes less pronounced. Furthermore, CPF outperforms all DPFs (since it does not use GM approximations and there are no unknown delays). Fig. 3 shows the dependence of ARMSE on the mean μ_τ of the processing-and-communication delays $\tau_{k,l}$. The performance degradation for increasing μ_τ is significant in the case of DPF-D but only small in the case of DPF-E. This again illustrates the good performance of the proposed scheme. We finally note that we did not observe a large RMSE variation across the sensors or simulation runs producing "lost tracks."

VII. Conclusion

We proposed a distributed sequential estimation scheme for asynchronous wireless sensor networks and a reduced-complexity implementation using particle filtering and Gaussian mixture approximations. Intersensor communications are limited to an exchange of Gaussian mixture parameters between neighbor sensors. We also presented an estimator of the delays resulting from processing and communication. Our simulation results demonstrate good performance of the proposed scheme and the benefits of delay estimation.

References

[1] T. Vercauteren and X. Wang, "Decentralized sigma-point information filters for target tracking in collaborative sensor networks," *IEEE Trans. Signal Process.*, vol. 53, pp. 2997–3009, Aug. 2005.

[2] 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.

[3] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Process. Mag.*, vol. 30, pp. 61–81, Jan. 2013.

[4] M. Vemula, J. Míguez, and A. Artés-Rodríguez, "A sequential Monte Carlo method for target tracking in an asynchronous wireless sensor network," in *Proc. WPNC-07*, Hannover, Germany, pp. 49–54, Mar. 2007.

[5] Y. Hu, Z. Duan, and D. Zhou, "Estimation fusion with general asynchronous multi-rate sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, pp. 2090–2102, Oct. 2010.

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

[7] H. Talebi and A. Hemmatyar, "Asynchronous track-to-track fusion by direct estimation of time of sample in sensor networks," *IEEE Sensors J.*, vol. 14, pp. 210–217, Jan. 2014.

[8] L. Geng, M. F. Bugallo, A. Athalye, and P. M. Djurić, "Indoor tracking with RFID systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 96–105, Feb. 2014.

[9] J. Beaudeau, M. F. Bugallo, and P. M. Djurić, "Target tracking with asynchronous measurements by a network of distributed mobile agents," in *Proc. IEEE ICASSP-12*, Kyoto, Japan, pp. 3857–3860, Mar. 2012.

[10] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ: Prentice Hall, 1993.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, Feb. 2002.

[12] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence," in *Proc. FUSION-12*, Singapore, pp. 1876–1883, Jul. 2012.

[13] W. J. Farrell III and C. Ganesh, "Generalized Chernoff fusion approximation for practical distributed data fusion," in *Proc. FUSION-09*, Seattle, WA, pp. 555–562, Jul. 2009.

[14] B. N. Oreshkin and M. J. Coates, "Asynchronous distributed particle filter via decentralized evaluation of Gaussian products," in *Proc. FUSION-10*, Edinburgh, UK, pp. 1–8, Jul. 2010.

[15] G. Battistelli, L. Chisci, and C. Fantacci, "Parallel consensus on likelihoods and priors for networked nonlinear filtering," *IEEE Signal Process. Letters*, vol. 21, pp. 787–791, Jul. 2014.

[16] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proc. IEEE*, vol. 95, pp. 899–924, May 2007.

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

[18] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, pp. 19–38, Sep. 2003.

[19] N. R. Ahmed and M. Campbell, "Fast consistent Chernoff fusion of Gaussian mixtures for ad hoc sensor networks," *IEEE Trans. Signal Process.*, vol. 60, pp. 6739–6745, Dec. 2012.

[20] S. J. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *Proc. FUSION-06*, Florence, Italy, pp. 1–8, Jul. 2006.