DISTRIBUTED DATA FUSION USING ITERATIVE COVARIANCE INTERSECTION

Ondrej Hlinka, Ondrej Slučiak, Franz Hlawatsch, and Markus Rupp

Institute of Telecommunications, Vienna University of Technology, Austria ({ohlinka, osluciak, fhlawats, mrupp}@nt.tuwien.ac.at)

ABSTRACT

We propose an iterative extension of the covariance intersection (CI) algorithm for distributed data fusion. Our iterative CI (ICI) algorithm is able to disseminate local information throughout the network. We show that the ICI algorithm converges asymptotically to a consensus across all network nodes. We furthermore apply the ICI algorithm to distributed sequential Bayesian estimation and propose an ICI-based distributed particle filter (DPF). This DPF allows for spatially correlated measurement noises with unknown cross-correlations and does not require knowledge of the network size. The performance of the proposed DPF is assessed experimentally for a target tracking problem.

***Index Terms*—** Distributed data fusion, covariance intersection, distributed estimation, distributed particle filter, sensor network.

1. INTRODUCTION

Contribution and relation to previous work. Distributed (decentralized) data fusion has important applications in sensor networks [1–4]. Frequently, the information to be fused throughout the network is in the form of local state estimates at the individual sensor nodes and corresponding estimated error covariance matrices. A well-known problem in this setting is the fact that the local estimates are often correlated with unknown cross-correlations. Ignoring these correlations may lead to "nonconservative" fused estimates with overconfident error covariance matrices [5, 6].

A popular approach to distributed data fusion is the covariance intersection (CI) algorithm, which conservatively fuses the local estimates for arbitrary cross-correlations [5–7]. In a network with communication links only between spatially close sensors, CI can only provide fusion of the associated local estimates. If the CI algorithm is employed as part of a sequential estimation scheme, local information can still be disseminated throughout the network, but only over many time steps (recursions) of the sequential estimation scheme.

Here, we introduce an *iterative CI (ICI) algorithm* that disseminates local information throughout the network. Although each sensor communicates only with its neighbors, the ICI algorithm converges asymptotically to a consensus on a global estimate that reflects the information of all sensors and is conservative. Our ICI algorithm is related to the average consensus-based fusion algorithm proposed in [8], with the difference that the weights used in our algorithm change in each iteration and depend on the fused quantities, but are independent of the network topology. This can lead to improved performance, as will be demonstrated using simulations.

We furthermore present an application of the ICI algorithm to distributed Bayesian filtering and propose an ICI-based distributed particle filter. In comparison to applying standard CI, the estimation performance is improved at the cost of additional intersensor communications. In contrast to state-of-the-art consensus-based distributed particle filters [9–12], the number of sensors need not be known. This is especially advantageous when the number of sensors changes over time. Furthermore, the local measurement noises are allowed to be correlated with unknown cross-correlations.

Paper outline. In Section 2, we formulate the distributed data fusion problem and review the CI algorithm. The proposed ICI algorithm is developed and discussed in Section 3. In Section 4, we apply the ICI algorithm to distributed Bayesian filtering and propose an ICI-based distributed particle filter algorithm. Finally, simulation results assessing the performance of the proposed distributed particle filter for a target tracking problem are presented in Section 5.

2. DISTRIBUTED DATA FUSION AND CI

We consider a sensor network composed of K sensors, where each sensor $k \in \{1,\dots,K\}$ is able to communicate with a set $\mathcal{N}_k \subseteq \{1,\dots,K\}\backslash\{k\}$ of neighboring sensors. The communication graph is assumed to be connected. Each sensor k calculates a local estimate $\hat{\mathbf{x}}_k \in \mathbb{R}^M$ of an unknown random state vector $\mathbf{x} \in \mathbb{R}^M$ and a corresponding estimated error covariance matrix $\hat{\mathbf{C}}_k \in \mathbb{R}^{M\times M}$. The local estimates $\hat{\mathbf{x}}_k$ are assumed unbiased, i.e., $\mathsf{E}\{\hat{\mathbf{x}}_k - \mathbf{x}\} = \mathbf{0}$, and "conservative" in the sense that $\hat{\mathbf{C}}_k - \mathbf{C}_k \geq \mathbf{0}$ [3,5]. Here, $\mathbf{C}_k$ is the true error covariance matrix associated with $\hat{\mathbf{x}}_k$, i.e., $\mathbf{C}_k = \mathsf{E}\big\{\big(\hat{\mathbf{x}}_k - \mathbf{x} - \mathsf{E}\{\hat{\mathbf{x}}_k - \mathbf{x}\}\big)\big(\hat{\mathbf{x}}_k - \mathbf{x} - \mathsf{E}\{\hat{\mathbf{x}}_k - \mathbf{x}\}\big)^\top\big\} = \mathsf{E}\{(\hat{\mathbf{x}}_k - \mathbf{x})(\hat{\mathbf{x}}_k - \mathbf{x})^\top\}$, and the notation $\mathbf{A} \geq \mathbf{0}$ expresses the fact that the square matrix $\mathbf{A}$ is positive semidefinite. The local estimates $\hat{\mathbf{x}}_k$ are allowed to be correlated with unknown cross-correlations.

Our goal is to fuse all local estimates $\hat{\mathbf{x}}_k$ and local covariances $\hat{\mathbf{C}}_k$ into a global estimate $\hat{\mathbf{x}}$ and a global covariance $\hat{\mathbf{C}}$ (here, $\hat{\mathbf{C}}$ is an estimate of the true global error covariance matrix associated with $\hat{\mathbf{x}}$, $\mathbf{C} = \mathsf{E}\{(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^\top\}$). The fusion should be performed in a distributed (decentralized) manner, using only local computations and communication with neighboring sensors, and in such a way that each sensor obtains the same global $\hat{\mathbf{x}}$ and $\hat{\mathbf{C}}$. Conservative fusion in the case of unknown cross-correlations of the $\hat{\mathbf{x}}_k$ is achieved by the CI algorithm [5, 6], which calculates $\hat{\mathbf{x}}$ and $\hat{\mathbf{C}}$ as

$$\hat{\mathbf{C}} = \left(\sum_{k=1}^{K} \omega_k \hat{\mathbf{C}}_k^{-1}\right)^{-1}, \quad \hat{\mathbf{x}} = \hat{\mathbf{C}} \sum_{k=1}^{K} \omega_k \hat{\mathbf{C}}_k^{-1} \hat{\mathbf{x}}_k \,. \tag{1}$$

Here, the weights ω_k satisfy $0 < \omega_k < 1$ and $\sum_{k=1}^{K} \omega_k = 1$ but are arbitrary otherwise. It is shown in [3, 5, 6] that the estimate $\hat{\mathbf{x}}$ is conservative, i.e., $\hat{\mathbf{C}} - \mathbf{C} \geq \mathbf{0}$, and unbiased, i.e., $\mathsf{E}\{\hat{\mathbf{x}} - \mathbf{x}\} = \mathbf{0}$.

To obtain $\hat{\mathbf{x}}$ and $\hat{\mathbf{C}}$ at each sensor using (1), a *fully* connected network is required or a suitable routing of the local estimates has to be employed. A straightforward adaptation of (1) to a (locally) connected network without routing is obtained by summing only over the extended neighborhood $\tilde{\mathcal{N}}_k \triangleq \mathcal{N}_k \cup \{k\}$ of each sensor k [13]. This means that sensor k obtains only partial, neighborhood-based estimates, denoted as $\hat{\mathbf{x}}_{\tilde{\mathcal{N}}_k}$ and $\hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k}$, which are given by

$$\hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k} = \left(\sum_{k'\in\tilde{\mathcal{N}}_k} \omega_{k,k'} \hat{\mathbf{C}}_{k'}^{-1}\right)^{-1}, \quad \hat{\mathbf{x}}_{\tilde{\mathcal{N}}_k} = \hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k} \sum_{k'\in\tilde{\mathcal{N}}_k} \omega_{k,k'} \hat{\mathbf{C}}_{k'}^{-1} \hat{\mathbf{x}}_{k'} \,. \tag{2}$$

Here, the weights $\omega_{k,k'}$ are now k-dependent and assumed to satisfy $0 < \omega_{k,k'} < 1$ and $\sum_{k'\in\tilde{\mathcal{N}}_k} \omega_{k,k'} = 1$.

This work was supported by the Austrian Science Fund (FWF) under grants S10603 and S10611 within the National Research Network SISE.

3. THE ITERATIVE CI ALGORITHM

Using the "local CI algorithm" (2), each sensor obtains a "partial" estimate $\hat{\mathbf{x}}_{\tilde{\mathcal{N}}_k}$ and covariance $\hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k}$ that reflect the local estimates $\hat{\mathbf{x}}_{k'}$ and covariances $\hat{\mathbf{C}}_{k'}$ only within the respective extended neighborhood, i.e., for $k' \in \tilde{\mathcal{N}}_k$. To overcome this limitation, we propose the iterative CI (ICI) algorithm. In each iteration of the ICI algorithm, each sensor performs a CI-based fusion of the partial estimates received from its neighbors. The result of this fusion is transmitted back to the neighbors and fused by them in the next iteration. Over the iterations, the information is disseminated throughout the network and, as we will show, a consensus across all sensors is reached asymptotically. Furthermore, since a CI-based fusion is used at each iteration, the estimates at each iteration are guaranteed to be unbiased and conservative.

Defining $\mathbf{\Gamma}_k \triangleq \hat{\mathbf{C}}_k^{-1}$, $\mathbf{\Gamma}_{\tilde{\mathcal{N}}_k} \triangleq \hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k}^{-1}$, $\boldsymbol{\gamma}_k \triangleq \hat{\mathbf{C}}_k^{-1}\hat{\mathbf{x}}_k$, and $\boldsymbol{\gamma}_{\tilde{\mathcal{N}}_k} \triangleq \hat{\mathbf{C}}_{\tilde{\mathcal{N}}_k}^{-1}\hat{\mathbf{x}}_{\tilde{\mathcal{N}}_k}$, the local CI algorithm (2) can be rewritten as

$$\mathbf{\Gamma}_{\tilde{\mathcal{N}}_k} = \sum_{k' \in \tilde{\mathcal{N}}_k} \omega_{k,k'} \mathbf{\Gamma}_{k'} , \quad \boldsymbol{\gamma}_{\tilde{\mathcal{N}}_k} = \sum_{k' \in \tilde{\mathcal{N}}_k} \omega_{k,k'} \boldsymbol{\gamma}_{k'} . \tag{3}$$

The ICI algorithm is an iterative variant of this formulation.

ITERATIVE CI (ICI) ALGORITHM

1. Local "states" $\boldsymbol{\gamma}_k^{(i)} \in \mathbb{R}^M$ and $\mathbf{\Gamma}_k^{(i)} \in \mathbb{R}^{M\times M}$ are initialized at each sensor k as $\boldsymbol{\gamma}_k^{(0)} = \hat{\mathbf{C}}_k^{-1}\hat{\mathbf{x}}_k$ and $\mathbf{\Gamma}_k^{(0)} = \hat{\mathbf{C}}_k^{-1}$.
2. For $i = 1, 2, \ldots, i_{\max}$, each sensor k performs the following steps:
   (a) The previous states $\boldsymbol{\gamma}_k^{(i-1)}$ and $\mathbf{\Gamma}_k^{(i-1)}$ are broadcast to all neighbors $k' \in \mathcal{N}_k$.
   (b) Weights $\omega_{k,k'}^{(i)}$ are calculated, e.g., as [7]
   $$\omega_{k,k'}^{(i)} = \frac{\operatorname{tr}\{(\mathbf{\Gamma}_{k'}^{(i-1)})^{-1}\}}{\sum_{l\in\tilde{\mathcal{N}}_k} \operatorname{tr}\{(\mathbf{\Gamma}_{l}^{(i-1)})^{-1}\}} , \quad k' \in \tilde{\mathcal{N}}_k ,$$
   where $\operatorname{tr}\{\cdot\}$ denotes the trace. Other weight definitions proposed in the CI literature—e.g., [6,14]—can be used as well. Note that the weights change with each iteration and depend on $\mathbf{\Gamma}_{k'}^{(i-1)}$ (possibly also on $\boldsymbol{\gamma}_k^{(i-1)}$). However, unlike, e.g., the frequently used Metropolis weights [8, 24], they do not depend on the network topology.
   (c) The local states are updated according to (cf. (3))
   $$\mathbf{\Gamma}_k^{(i)} = \sum_{k' \in \tilde{\mathcal{N}}_k} \omega_{k,k'}^{(i)} \mathbf{\Gamma}_{k'}^{(i-1)}, \quad \boldsymbol{\gamma}_k^{(i)} = \sum_{k' \in \tilde{\mathcal{N}}_k} \omega_{k,k'}^{(i)} \boldsymbol{\gamma}_{k'}^{(i-1)}. \tag{4}$$
3. Estimates of $\mathbf{x}$ and $\mathbf{C}$ are derived from $\boldsymbol{\gamma}_k^{(i_{\max})}$ and $\mathbf{\Gamma}_k^{(i_{\max})}$ as
   $$\hat{\mathbf{C}}_k^{(i_{\max})} = \big(\mathbf{\Gamma}_k^{(i_{\max})}\big)^{-1}, \quad \hat{\mathbf{x}}_k^{(i_{\max})} = \hat{\mathbf{C}}_k^{(i_{\max})} \boldsymbol{\gamma}_k^{(i_{\max})} . \tag{5}$$

In Step 2, each sensor broadcasts $i_{\max}\,[M + M(M+1)/2]$ real numbers to its neighbors.

The following results are proved in the Appendix.

Proposition 1 The estimates $\hat{\mathbf{x}}_k^{(i_{\max})}$ and $\hat{\mathbf{C}}_k^{(i_{\max})}$ in (5) satisfy the following properties:

(i) They are unbiased, i.e., $\mathsf{E}\big\{\hat{\mathbf{x}}_k^{(i_{\max})} - \mathbf{x}\big\} = \mathbf{0}$ for all $i_{\max}$.

(ii) They are conservative, i.e., $\hat{\mathbf{C}}_k^{(i_{\max})} - \mathbf{C}_k^{(i_{\max})} \geq \mathbf{0}$ for all $i_{\max}$, where $\mathbf{C}_k^{(i_{\max})} \triangleq \mathsf{E}\big\{\big(\hat{\mathbf{x}}_k^{(i_{\max})} - \mathbf{x}\big)\big(\hat{\mathbf{x}}_k^{(i_{\max})} - \mathbf{x}\big)^\top\big\}$.

(iii) For $i_{\max} \to \infty$, they become equal for all $k \in \{1, \ldots, K\}$, i.e., $\lim_{i_{\max}\to\infty} \hat{\mathbf{x}}_k^{(i_{\max})} = \hat{\mathbf{x}}^{(\infty)}$ and $\lim_{i_{\max}\to\infty} \hat{\mathbf{C}}_k^{(i_{\max})} = \hat{\mathbf{C}}^{(\infty)}$.

Properties (i) and (ii) are consequences of, respectively, the unbiasedness and conservativeness assumptions made in Section 2 (i.e., $\mathsf{E}\{\hat{\mathbf{x}}_k - \mathbf{x}\} = \mathbf{0}$ and $\hat{\mathbf{C}}_k - \mathbf{C}_k \geq \mathbf{0}$). Property (iii) means that, after convergence, all local states in (4) have reached a consensus across all sensors. In contrast to the result of an average consensus algorithm [15] or the fusion algorithm in [8], the asymptotic values are not arithmetic averages of the initial states in general. Furthermore, the estimates in (5) are generally different from the estimates obtained using the "standard CI algorithm" (1) (recall that (1) presupposes a fully connected network or the use of routing). However, in Section 5, we will demonstrate experimentally that use of the two algorithms within a distributed particle filter leads to a similar estimation performance. We also note that for $i_{\max} = 1$, the ICI algorithm is identical to the "local CI algorithm" (2). In what follows, we assume that $i_{\max}$ is sufficiently large, so that any differences between $\hat{\mathbf{x}}_k^{(i_{\max})}$, $\hat{\mathbf{C}}_k^{(i_{\max})}$ at different sensors k are negligible, and we will thus denote these quantities simply as $\hat{\mathbf{x}}$ and $\hat{\mathbf{C}}$, without the index k.

4. APPLICATION TO DISTRIBUTED BAYESIAN FILTERING

We will now show how the ICI algorithm can be used to obtain a distributed sequential Bayesian estimation (filtering) scheme and, in particular, a distributed particle filter. We consider a random, time-varying state vector $\mathbf{x}_n \in \mathbb{R}^M$ that evolves according to

$$\mathbf{x}_n = \mathbf{g}_n(\mathbf{x}_{n-1}, \mathbf{u}_n) , \quad n \in \{1, 2, \ldots\}, \tag{6}$$

where $\mathbf{g}_n(\cdot\,,\cdot)$ is a generally nonlinear function and $\mathbf{u}_n$ is white driving noise with a known probability density function (pdf) $f(\mathbf{u}_n)$. At time n, $\mathbf{x}_n$ is sensed by K sensors according to the measurement models

$$\mathbf{z}_{n,k} = \mathbf{h}_{n,k}(\mathbf{x}_n, \mathbf{v}_{n,k}) , \quad k \in \{1, \ldots, K\}. \tag{7}$$

Here, $\mathbf{z}_{n,k} \in \mathbb{R}^{N_{n,k}}$ is the measurement of sensor k, $\mathbf{h}_{n,k}(\cdot\,,\cdot)$ is a generally nonlinear function, and $\mathbf{v}_{n,k}$ is measurement noise with a known pdf $f(\mathbf{v}_{n,k})$. We assume that (i) $\mathbf{v}_{n,k}$ and $\mathbf{v}_{n',k'}$ are independent unless $n = n'$; (ii) the initial state $\mathbf{x}_0$ and the sequences $\mathbf{u}_n$ and $\mathbf{v}_{n,k}$ are all independent; and (iii) sensor k knows $\mathbf{g}_n(\cdot\,,\cdot)$ and $\mathbf{h}_{n,k}(\cdot\,,\cdot)$ but not $\mathbf{h}_{n,k'}(\cdot\,,\cdot)$ for $k' \neq k$. The measurement noises at different sensors, $\mathbf{v}_{n,k}$ and $\mathbf{v}_{n,k'}$ for $k \neq k'$, are allowed to be correlated with unknown cross-correlations. We denote by $\mathbf{z}_n \triangleq (\mathbf{z}_{n,1}^\top \cdots \mathbf{z}_{n,K}^\top)^\top$ the vector containing the measurements of all sensors at time n. Equations (6) and (7) together with our statistical assumptions determine the state-transition pdf $f(\mathbf{x}_n|\mathbf{x}_{n-1})$, the local likelihood function $f(\mathbf{z}_{n,k}|\mathbf{x}_n)$, and the global (all-sensors) likelihood function $f(\mathbf{z}_n|\mathbf{x}_n)$.

4.1. ICI-based Distributed Bayesian Filtering

Our goal is to estimate the state $\mathbf{x}_n$ from $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^\top \cdots \mathbf{z}_n^\top)^\top$, i.e., from the measurements of all sensors up to time n, using a distributed (decentralized) scheme in which each sensor obtains an estimate reflecting the measurements of all sensors. We consider the minimum mean-square error (MMSE) estimator [16]

$$\hat{\mathbf{x}}_n^{\text{MMSE}} \triangleq \mathrm{E}\{\mathbf{x}_n|\mathbf{z}_{1:n}\} = \int_{\mathbb{R}^M} \mathbf{x}_n\, f(\mathbf{x}_n|\mathbf{z}_{1:n})\, d\mathbf{x}_n . \tag{8}$$

The posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ in (8) can be calculated sequentially from the previous posterior $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ and the global likelihood function $f(\mathbf{z}_n|\mathbf{x}_n)$ [17]. However, straightforward distributed calculation of $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ presupposes that all sensor measurements and the global likelihood function are available at each sensor.

Here, we propose the following alternative approach. First, each sensor k computes its own *local* posterior pdf

$$f(\mathbf{x}_n|\mathbf{z}_{1:n-1},\mathbf{z}_{n,k}) \propto f(\mathbf{z}_{n,k}|\mathbf{x}_n)\int f(\mathbf{x}_n|\mathbf{x}_{n-1})\,f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})\,d\mathbf{x}_{n-1}\,. \quad (9)$$

This merely presupposes that sensor k knows its own measurement $\mathbf{z}_{n,k}$ and local likelihood function $f(\mathbf{z}_{n,k}|\mathbf{x}_n)$, and that it obtained the previous global posterior $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ at time $n-1$ (as will be discussed presently). Then, for distributed data fusion, the local posterior (9) is approximated by a Gaussian pdf, i.e.,

$$f(\mathbf{x}_n|\mathbf{z}_{1:n-1},\mathbf{z}_{n,k}) \approx \mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}_{n,k},\hat{\mathbf{C}}_{n,k})\,, \quad (10)$$

where $\hat{\mathbf{x}}_{n,k}$ and $\hat{\mathbf{C}}_{n,k}$ are, respectively, the mean vector and covariance matrix of $f(\mathbf{x}_n|\mathbf{z}_{1:n-1},\mathbf{z}_{n,k})$. Next, the local means $\hat{\mathbf{x}}_{n,k}$ and covariances $\hat{\mathbf{C}}_{n,k}$ of all sensors k are fused in a distributed way using the ICI algorithm. The resulting global mean $\hat{\mathbf{x}}_n$ and covariance $\hat{\mathbf{C}}_n$ establish a Gaussian approximation of the global posterior, i.e.,

$$f(\mathbf{x}_n|\mathbf{z}_{1:n}) \approx \mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}_n,\hat{\mathbf{C}}_n)\,. \quad (11)$$

This is used at the next time $n+1$ to compute the local posterior according to (9). Furthermore, the global mean $\hat{\mathbf{x}}_n$ provides an approximation of the global MMSE estimate (8).

4.2. An ICI-based Distributed Particle Filter

For a general nonlinear/non-Gaussian system model, straightforward evaluation of (9) is typically infeasible. A computationally feasible approximation can be obtained by a particle filter (PF), which represents the local posterior by samples (particles) and associated weights [18–20]. Next, we present a distributed particle implementation of the distributed Bayesian filter described in Section 4.1. In this distributed PF (DPF), each sensor runs a local PF that computes a particle representation of the local posterior (9). For a tractable distributed fusion of local posteriors, a Gaussian representation (10) is calculated at each sensor from the local particles. A Gaussian representation (11) of the global posterior is then obtained by using the ICI algorithm to fuse the local Gaussian representations.

ICI-BASED DISTRIBUTED PF ALGORITHM

The local PF at sensor k is initialized at time $n=0$ with J particles $\{\mathbf{x}_{0,k}^{(j)}\}_{j=1}^{J}$ randomly drawn from a prior pdf $f(\mathbf{x}_0)$.

At time $n\geq 1$, J particles $\{\mathbf{x}_{n-1,k}^{(j)}\}_{j=1}^{J}$ are available at sensor k from the previous recursion (at time $n-1$). The local PF at sensor k then updates these particles by performing the following steps:

1. For each particle $\mathbf{x}_{n-1,k}^{(j)}$, a new "predicted" particle $\bar{\mathbf{x}}_{n,k}^{(j)}$ is drawn from $f(\mathbf{x}_n|\mathbf{x}_{n-1,k}^{(j)}) \equiv f(\mathbf{x}_n|\mathbf{x}_{n-1})\big|_{\mathbf{x}_{n-1}=\mathbf{x}_{n-1,k}^{(j)}}$.
2. Weights $\eta_{n,k}^{(j)}$ associated with the particles $\bar{\mathbf{x}}_{n,k}^{(j)}$ are calculated by setting $\tilde{\eta}_{n,k}^{(j)} = f(\mathbf{z}_{n,k}|\bar{\mathbf{x}}_{n,k}^{(j)})$ and performing a normalization, i.e., $\eta_{n,k}^{(j)} = \tilde{\eta}_{n,k}^{(j)}/\sum_{j'=1}^{J}\tilde{\eta}_{n,k}^{(j')}$. The set $\{(\bar{\mathbf{x}}_{n,k}^{(j)},\eta_{n,k}^{(j)})\}_{j=1}^{J}$ provides a particle representation of the current local posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n-1},\mathbf{z}_{n,k})$.
3. From $\{(\bar{\mathbf{x}}_{n,k}^{(j)},\eta_{n,k}^{(j)})\}_{j=1}^{J}$, an approximation of the local posterior mean is computed as $\hat{\mathbf{x}}_{n,k} = \sum_{j=1}^{J}\eta_{n,k}^{(j)}\bar{\mathbf{x}}_{n,k}^{(j)}$, and an associated error covariance matrix (posterior covariance) is estimated as $\hat{\mathbf{C}}_{n,k} = \sum_{j=1}^{J}\eta_{n,k}^{(j)}(\bar{\mathbf{x}}_{n,k}^{(j)}-\hat{\mathbf{x}}_{n,k})(\bar{\mathbf{x}}_{n,k}^{(j)}-\hat{\mathbf{x}}_{n,k})^{\top} = \sum_{j=1}^{J}\eta_{n,k}^{(j)}\bar{\mathbf{x}}_{n,k}^{(j)}\bar{\mathbf{x}}_{n,k}^{(j)\top} - \hat{\mathbf{x}}_{n,k}\hat{\mathbf{x}}_{n,k}^{\top}$.
4. The ICI algorithm is executed, using the estimates $\hat{\mathbf{x}}_{n,k}$ and $\hat{\mathbf{C}}_{n,k}$ for $k\in\{1,\ldots,K\}$ as its input. As a result of the ICI algorithm, each sensor obtains global quantities $\hat{\mathbf{x}}_n$ and $\hat{\mathbf{C}}_n$, which establish a Gaussian approximation $\mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}_n,\hat{\mathbf{C}}_n)$ of the global posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ (cf. (11)). Furthermore, $\hat{\mathbf{x}}_n$ is an approximation of the global MMSE estimate (8).
5. J particles $\{\mathbf{x}_{n,k}^{(j)}\}_{j=1}^{J}$ are sampled from $\mathcal{N}(\mathbf{x}_n;\hat{\mathbf{x}}_n,\hat{\mathbf{C}}_n)$.

Note that only Step 4 requires communication between (neighboring) sensors. The total count of real numbers that are broadcast by each sensor to its neighbors at time n is $i_{\max}[M+M(M+1)/2]$, where, as before, $i_{\max}$ is the number of CI iterations and M is the dimension of $\mathbf{x}_n$. All other steps are performed locally at sensor k, using only locally available information.

In contrast to most consensus-based DPFs (e.g., [9–12]), the ICI-based DPF presented above does not require knowledge of the number of sensors in the network. This makes it particularly suitable for networks with a time-varying number of sensors. With conventional consensus-based DPFs, the number of sensors would have to be estimated in each time step. This may introduce significant additional latency, especially in large networks, and an incorrect estimate of the number of sensors can degrade the performance of the DPF. Another advantage of our ICI-based DPF is the fact that the local measurement noises are allowed to be correlated with unknown cross-correlations. Most consensus-based DPF schemes assume uncorrelated measurement noises, exceptions including [21, 22].

5. NUMERICAL RESULTS

We consider a target tracking application in which the state vector $\mathbf{x}_n = (x_n\ y_n\ \dot{x}_n\ \dot{y}_n)^{\top}$ represents the two-dimensional position and velocity of a single target in the x-y plane. The state vector evolves according to (cf. (6))

$$\mathbf{x}_n = \mathbf{G}\mathbf{x}_{n-1} + \mathbf{H}\mathbf{u}_n\,, \quad n\in\{1,2,\ldots\}\,,$$

where the matrices $\mathbf{G}\in\mathbb{R}^{4\times 4}$ and $\mathbf{H}\in\mathbb{R}^{4\times 2}$ are chosen as in [9] and the driving noise vectors $\mathbf{u}_n\in\mathbb{R}^2$ are independent and identically distributed according to $\mathcal{N}(\mathbf{0}_2,\sigma_u^2\mathbf{I}_2)$ with $\sigma_u^2 = 0.01$. The network consists of $K=25$ sensors, which are deployed on a jittered grid within a square of size $d_a\times d_a$ with $d_a=40$. Each sensor communicates with other sensors within a radius of $d_c=11$. The (scalar) measurement of sensor k is given by (cf. (7))

$$z_{n,k} = \frac{A}{\|\boldsymbol{\rho}(\mathbf{x}_n)-\boldsymbol{\xi}_k\|^2} + v_{n,k}\,,$$

where $\boldsymbol{\rho}(\mathbf{x}_n)\triangleq(x_n\ y_n)^{\top}$ is the position of the target, $\boldsymbol{\xi}_k$ is the position of sensor k, and $A=10$ is the amplitude of an acoustic or electromagnetic signal emitted by the target. Unless specified otherwise, the all-sensors measurement noise vector $\mathbf{v}_n \triangleq (v_{n,1}\cdots v_{n,25})^{\top}$ is distributed as $\mathbf{v}_n\sim\mathcal{N}(\mathbf{0},\mathbf{Q}^{-1})$ and independent for different times n. Here, the entries of the precision matrix $\mathbf{Q}$ are chosen as [21, 23]

$$Q_{k,k'} = \begin{cases} q\,, & k=k' \\ -q'(d_a-\|\boldsymbol{\xi}_k-\boldsymbol{\xi}_{k'}\|)\,, & k\neq k', \end{cases}$$

with $q=10^5$ and $q'=180$. Thus, the measurement noise $v_{n,k}$ is correlated across the sensors k.

We simulated the following PF methods: (i) the proposed ICI-based DPF (abbreviated ICI-DPF); (ii) a DPF that uses the fusion rule [8] with Metropolis weights [24] (abbreviated M-DPF); (iii) a state-of-the-art consensus-based DPF (abbreviated C-DPF) that uses an average consensus algorithm to calculate a Gaussian approximation of the global posterior from Gaussian approximations of local

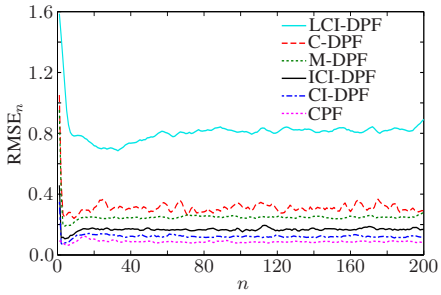

Fig. 1. RMSE$_n$ versus time n for correlated measurement noises and $i_{\max} = 10$ CI or consensus iterations.

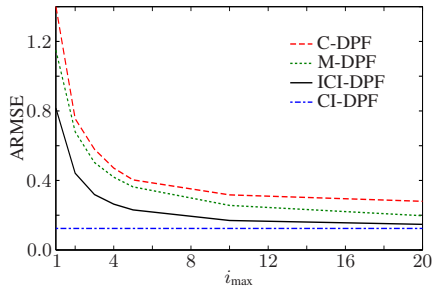

Fig. 2. ARMSE versus number of CI or consensus iterations $i_{\max}$ for correlated measurement noises.

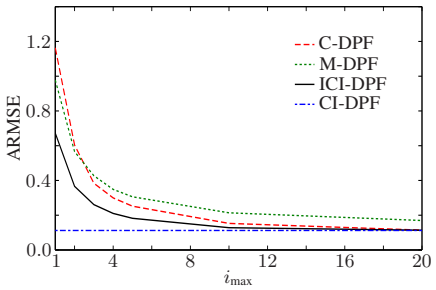

Fig. 3. ARMSE versus number of CI or consensus iterations $i_{\max}$ for uncorrelated measurement noises.

posteriors [9, 10]; (iv) a DPF that uses the standard CI algorithm in (1) to fuse *all* local estimates in a single step (abbreviated CI-DPF; the weights were chosen as suggested in [7]); (v) a DPF using the local CI algorithm in (2) (abbreviated LCI-DPF; this equals ICI-DPF with $i_{\max} = 1$); and (vi) a centralized PF (CPF) that transmits all measurements to a fusion center for processing. The communication requirements of ICI-DPF, M-DPF, and C-DPF are exactly equal: in each iteration, each sensor broadcasts to its neighbors a mean vector and a covariance matrix. Note that CI-DPF presupposes a fully connected network or routing of estimates from each sensor to all other sensors, which is often impractical.

The number of particles at each sensor (for the DPFs) and at the fusion center (for the CPF) is $J = 5000$. The DPFs know only the diagonal entries of the noise precision matrix $\mathbf{Q}$ (i.e., q), whereas the CPF knows the entire $\mathbf{Q}$. As a performance measure, we use the root-mean-square error of the estimate of $\boldsymbol{\rho}(\mathbf{x}_n)$, denoted RMSE$_n$, which is computed as the square root of the average of the squared position estimation error at time n over all sensors and 1000 simulation runs. We also compute the *average* RMSE (ARMSE) by averaging RMSE$_n^2$ over all 200 simulated time instants n and taking the square root of the result.

Fig. 1 shows the evolution of RMSE$_n$. Here, ICI-DPF and M-DPF use $i_{\max} = 10$ CI iterations and C-DPF uses 10 average consensus iterations. We observe that ICI-DPF outperforms C-DPF, M-DPF, and LCI-DPF and performs almost as well as CI-DPF and CPF.

Fig. 2 shows the ARMSE versus the number of CI iterations used by ICI-DPF and M-DPF or the number of average consensus iterations used by C-DPF, both denoted by $i_{\max}$. As expected, the ARMSE decreases with growing $i_{\max}$. The ARMSE of CI-DPF (which does not depend on $i_{\max}$ because CI-DPF does not employ ICI or average consensus) is also shown as a reference. For growing $i_{\max}$, the ARMSE of ICI-DPF comes close to that of CI-DPF. Furthermore, ICI-DPF outperforms both C-DPF and M-DPF.

Fig. 3 shows the ARMSE versus $i_{\max}$ for the case of uncorrelated measurement noises. The local measurement noises $v_{n,k}$ are Gaussian with variance $\sigma_v^2 = 1/q = 0.00001$. Again, ICI-DPF outperforms C-DPF and M-DPF, and its ARMSE comes close to that of CI-DPF for large $i_{\max}$. However, C-DPF now performs better than previously, and for higher $i_{\max}$ it approaches the performance of ICI-DPF. This result is not unexpected because C-DPF assumes uncorrelated measurement noises.

6. CONCLUSION

The proposed iterative covariance intersection (ICI) algorithm for distributed data fusion disseminates local information throughout the network and provides each sensor with a global estimate. The ICI algorithm uses only local communication between neighboring sensors; a fully connected network or routing is not required. Convergence of the ICI algorithm to a consensus across all sensors is guaranteed asymptotically. We applied the ICI algorithm to distributed sequential Bayesian estimation and, in particular, proposed an ICI-based distributed particle filter (DPF). In contrast to consensus-based DPFs, our DPF does not require knowledge of the number of sensors; it is hence well suited to networks with a time-varying number of sensors. Furthermore, the proposed DPF allows for correlated sensor measurement noises with unknown cross-correlations. Simulation results for a target tracking problem demonstrated the good performance of the proposed ICI-based DPF.

APPENDIX: PROOF OF PROPOSITION 1

We define the weighted adjacency matrix [25] $\mathbf{W}^{(i)} \in \mathbb{R}^{K\times K}$ by $[\mathbf{W}^{(i)}]_{k,k'} = \omega_{k,k'}^{(i)}$ for $k' \in \tilde{\mathcal{N}}_k$ and $[\mathbf{W}^{(i)}]_{k,k'} = 0$ for $k' \in \{1,\dots,K\} \setminus \tilde{\mathcal{N}}_k$. Furthermore, $\widetilde{\mathbf{W}}^{(i)} \triangleq \prod_{j=1}^{i} \mathbf{W}^{(j)}$.

(i) The global estimates $\hat{\mathbf{x}}_k^{(i_{\max})}$ are linear functions of the local estimates $\hat{\mathbf{x}}_{k'}$, $k' \in \{1,\dots,K\}$:

$$\hat{\mathbf{x}}_k^{(i_{\max})} = \hat{\mathbf{C}}_k^{(i_{\max})} \sum_{k'=1}^{K} \left[\widetilde{\mathbf{W}}^{(i_{\max})}\right]_{k,k'} \hat{\mathbf{C}}_{k'}^{-1} \hat{\mathbf{x}}_{k'} \,.$$

(Note that $\hat{\mathbf{C}}_k^{(i_{\max})}$ and $\widetilde{\mathbf{W}}^{(i_{\max})}$ do not depend on the $\hat{\mathbf{x}}_{k'}$.) Since the $\hat{\mathbf{x}}_{k'}$ are unbiased by our assumption in Section 2 and a linear transformation preserves unbiasedness [3, 16], the global estimates $\hat{\mathbf{x}}_k^{(i_{\max})}$ are themselves unbiased.

(ii) The global estimates $\hat{\mathbf{x}}_k^{(i_{\max})}$ are derived from the local estimates $\hat{\mathbf{x}}_{k'}$, $k' \in \{1,\dots,K\}$ by a sequence of successive CI updates (4). The local estimates $\hat{\mathbf{x}}_{k'}$ are conservative by our assumption in Section 2. It is known [5–7] that a CI update preserves conservativeness. Hence, this is also true for any sequence of successive CI updates. It thus follows that the global estimates $\hat{\mathbf{x}}_k^{(i_{\max})}$ are conservative.

(iii) For $j \in \{1,\dots,M\}$ arbitrary but fixed, let $\xi_k^{(i)} \triangleq [\boldsymbol{\gamma}_k^{(i)}]_j$ and $\boldsymbol{\xi}^{(i)} \triangleq (\xi_1^{(i)} \cdots \xi_K^{(i)})^\top$. From the second equation in (4), we obtain

$$\boldsymbol{\xi}^{(i)} = \mathbf{W}^{(i)} \boldsymbol{\xi}^{(i-1)} = \widetilde{\mathbf{W}}^{(i)} \boldsymbol{\xi}^{(0)}.$$

The matrix sequence $\mathbf{W}^{(i)}$ satisfies the conditions in [26, Assumption 2.1], and thus $\widetilde{\mathbf{W}}^{(\infty)} = \lim_{i\to\infty} \widetilde{\mathbf{W}}^{(i)}$ exists and has all rows identical. Hence, $\boldsymbol{\xi}^{(\infty)} = \widetilde{\mathbf{W}}^{(\infty)} \boldsymbol{\xi}^{(0)}$ is a vector with identical components, i.e., for $j \in \{1,\dots,M\}$ fixed, each sensor k obtains the same value of $[\boldsymbol{\gamma}_k^{(\infty)}]_j$. Since the considered j was arbitrary, we conclude that $\boldsymbol{\gamma}_k^{(\infty)}$ is identical for all k, i.e., $\boldsymbol{\gamma}_k^{(\infty)} = \boldsymbol{\gamma}^{(\infty)}$. A similar argument (now setting $\xi_k^{(i)} \triangleq [\boldsymbol{\Gamma}_k^{(i)}]_{j,j'}$ and using the first equation in (4)) shows that also $\boldsymbol{\Gamma}_k^{(\infty)} = \boldsymbol{\Gamma}^{(\infty)}$ for all k. From (5), we then conclude that $\hat{\mathbf{C}}_k^{(\infty)} = \hat{\mathbf{C}}^{(\infty)}$ and $\hat{\mathbf{x}}_k^{(\infty)} = \hat{\mathbf{x}}^{(\infty)}$ for all k.

7. REFERENCES

[1] M. E. Liggins, D. L. Hall, and J. Llinas, *Handbook of Multisensor Data Fusion: Theory and Practice*. Boca Raton, FL: CRC Press, 2nd ed., 2009.

[2] S. J. Julier, T. Bailey, and J. K. Uhlmann, "Using exponential mixture models for suboptimal distributed data fusion," in *Proc. IEEE NSSPW-06*, Cambridge, UK, pp. 160–163, Sep. 2006.

[3] Y. Wang and X. R. Li, "Distributed estimation fusion with unavailable cross-correlation," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 48, pp. 259–278, Jan. 2012.

[4] Z. Weng and P. M. Djurić, "Data fusion based on convex optimization," in *Proc. IEEE ICASSP-13*, Vancouver, BC, Canada, pp. 6605–6609, May 2013.

[5] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. ACC-97*, Albuquerque, NM, pp. 2369–2373, Jun. 1997.

[6] S. J. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection," in *Handbook of Multisensor Data Fusion: Theory and Practice* (M. E. Liggins, D. L. Hall, and J. Llinas, eds.), pp. 319–344, Boca Raton, FL: CRC Press, 2nd ed., 2009.

[7] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proc. FUSION-02*, Annapolis, MD, pp. 901–904, Jul. 2002.

[8] G. Battistelli, L. Chisci, S. Morrocchi, and F. Papi, "An information-theoretic approach to distributed state estimation," in *Proc. 18th IFAC World Congr.*, Milan, Italy, pp. 12477–12482, Aug. 2011.

[9] 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.

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

[11] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. Signal Process.*, vol. 59, pp. 4122–4138, Sep. 2011.

[12] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, pp. 4334–4349, Aug. 2012.

[13] J. Hu, L. Xie, and C. Zhang, "Diffusion Kalman filtering based on covariance intersection," *IEEE Trans. Signal Process.*, vol. 60, pp. 891–902, Feb. 2012.

[14] D. Fränken and A. Hüpper, "Improved fast covariance intersection for distributed data fusion," in *Proc. FUSION-05*, Philadelphia, PA, pp. 154–160, Jul. 2005.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, pp. 215–233, Jan. 2007.

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

[17] H. Tanizaki, *Nonlinear Filters: Estimation and Applications*. Berlin, Germany: Springer, 1996.

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

[19] 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.

[20] 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.

[21] O. Hlinka and F. Hlawatsch, "Distributed particle filtering in the presence of mutually correlated sensor noises," in *Proc. IEEE ICASSP-13*, Vancouver, BC, Canada, pp. 6269–6273, May 2013.

[22] M. Moldaschl, W. N. Gansterer, O. Hlinka, F. Meyer, and F. Hlawatsch, "Distributed decorrelation in sensor networks with application to distributed particle filtering," in *Proc. IEEE ICASSP-14*, Florence, Italy, May 2014.

[23] R. G. Baraniuk, V. Delouille, and R. Neelamani, "Robust distributed estimation using the embedded subgraphs algorithm," *IEEE Trans. Signal Process.*, vol. 54, pp. 2998–3010, Aug. 2006.

[24] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IEEE/ACM IPSN-05*, Los Angeles, CA, pp. 63–70, Apr. 2005.

[25] A. E. Brouwer and W. H. Haemers, *Spectra of Graphs*. New York, NY: Springer, 2012.

[26] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, pp. 33–55, Feb. 2009.