 in Proc. 45th Asilomar Conf. Signals, Systems, Computers, Pacific Grove, CA, Nov. 2011

Sequential Likelihood Consensus and Its Application to Distributed Particle Filtering with Reduced Communications and Latency

Ondrej Slučiak[a], *Ondrej Hlinka*[a], *Markus Rupp*[a], *Franz Hlawatsch*[a], and *Petar M. Djurić*[b]

[a]Institute of Telecommunications, Vienna University of Technology (osluciak@nt.tuwien.ac.at)
[b]Department of Electrical and Computer Engineering, Stony Brook University (djuric@ece.sunysb.edu)

***Abstract*—We propose a *sequential likelihood consensus* (SLC) for a distributed, sequential computation of the joint (all-sensors) likelihood function (JLF) in a wireless sensor network. The SLC is based on a novel dynamic consensus algorithm, of which only a single iteration is performed per time step. We demonstrate the application of the SLC in a distributed particle filter with low communication requirements and low latency. Because the JLF is available at each sensor, the local particle filters at the individual sensors take into account the measurements of all sensors. The performance of the proposed distributed particle filter is assessed for a target tracking problem.**

***Index Terms*—Likelihood consensus, distributed particle filter, distributed estimation, target tracking, wireless sensor network.**

I. Introduction

For distributed inference in energy-constrained wireless sensor networks without a fusion center, it is often important to keep the amount of inter-sensor communications and the latency as small as possible (e.g., [1], [2]). In this paper, we consider decentralized sequential state estimation in a wireless sensor network, using only *local* processing and *low-rate*, *local* inter-sensor communications. To obtain a *global* estimate at each sensor, the sensors must know the joint (all-sensors) likelihood function (JLF). In our previous work [3]–[6], we proposed the likelihood consensus (LC) method for a distributed approximate computation of the JLF. Here, we propose an extension of the LC method for computing an approximation of the JLF sequentially. We refer to this algorithm as *sequential LC* (SLC). The SLC is based on a novel dynamic consensus algorithm, of which only a single iteration is performed per time step. We then apply the SLC to obtain a reduced-communications, reduced-latency distributed particle filter (PF).

This paper is structured as follows. In Section II, we describe the system model and review the basics of sequential Bayesian estimation. In Section III, we present a dynamic consensus algorithm and the SLC algorithm. An SLC-based distributed PF is discussed in Section IV. In Section V, we present simulation results demonstrating the performance of our distributed PF for a target tracking problem.

This work was supported by the Austrian Science Fund (FWF) under Awards S10611 and S10603 within the National Research Network SISE, by the NSF under Award CCF-1018323, and by the ONR under Award N00014-09-1-1154.

II. System Model and Sequential Bayesian Estimation

We consider a wireless sensor network consisting of K sensors that communicate with their neighbors within a given radius. At any given discrete time n, each sensor estimates a global state vector $\mathbf{x}_n = (x_{n,1} \cdots x_{n,M})^\top \in \mathbb{R}^M$ based on the measurements of all sensors. The state evolves according to the state-transition probability density function (pdf) $f(\mathbf{x}_n|\mathbf{x}_{n-1})$. At time n, the kth sensor ($k \in \{1,\dots,K\}$) acquires a measurement vector $\mathbf{z}_{n,k} \in \mathbb{R}^{N_{n,k}}$. The relationship between $\mathbf{z}_{n,k}$ and $\mathbf{x}_n$ is described by the *local likelihood function* $f(\mathbf{z}_{n,k}|\mathbf{x}_n)$, and the relationship between the all-sensors measurement vector $\mathbf{z}_n \triangleq (\mathbf{z}_{n,1}^\top \cdots \mathbf{z}_{n,K}^\top)^\top$ and $\mathbf{x}_n$ is given by the JLF $f(\mathbf{z}_n|\mathbf{x}_n)$. All $\mathbf{z}_{n,k}$ are assumed conditionally independent given $\mathbf{x}_n$, i.e.,

$$f(\mathbf{z}_n|\mathbf{x}_n) = \prod_{k=1}^{K} f(\mathbf{z}_{n,k}|\mathbf{x}_n)\,. \tag{1}$$

Let $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^\top \cdots \mathbf{z}_n^\top)^\top$ denote the vector collecting the measurements of all sensors from time 1 to time n. We will use the following assumptions. First, the current state $\mathbf{x}_n$ is conditionally independent of all past measurements, $\mathbf{z}_{1:n-1}$, given the previous state $\mathbf{x}_{n-1}$, i.e.,

$$f(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z}_{1:n-1}) = f(\mathbf{x}_n|\mathbf{x}_{n-1})\,. \tag{2}$$

Second, the current measurement $\mathbf{z}_n$ is conditionally independent of all past measurements given the current state $\mathbf{x}_n$, i.e.,

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

We also assume that sensor k knows $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ and $f(\mathbf{z}_{n,k}|\mathbf{x}_n)$ for all n, as well as the pdf $f(\mathbf{x}_0)$ of the initial state $\mathbf{x}_0$; however, sensor k cannot compute the local likelihood functions of the other sensors, i.e., $f(\mathbf{z}_{n,k'}|\mathbf{x}_n)$ for $k' \neq k$, because it does not know the measurements of these sensors.

Our goal is to perform sequential Bayesian estimation of the state $\mathbf{x}_n$ from the past and present measurements of all

sensors, $\mathbf{z}_{1:n}$. For this task, we consider the minimum mean-square error (MMSE) estimator [7],

$$\hat{\mathbf{x}}_n^{\mathrm{MMSE}} \triangleq \mathrm{E}\{\mathbf{x}_n|\mathbf{z}_{1:n}\} = \int \mathbf{x}_n f(\mathbf{x}_n|\mathbf{z}_{1:n})\, d\mathbf{x}_n\,, \qquad (4)$$

which is implemented at each sensor. Using the state-transition pdf $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ as well as (2) and (3), the posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ involved in (4) can be calculated sequentially from the previous posterior $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1})$ and the JLF $f(\mathbf{z}_n|\mathbf{x}_n)$ by means of the following temporal recursion [8]:

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

However, in the nonlinear/non-Gaussian case considered, the computational complexity of sequential MMSE state estimation as given by (4) and (5) is typically prohibitive. A computationally feasible approximation is provided by the PF [9]. In a PF, the posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is represented by a set of samples (or particles) $\mathbf{x}_n^{(j)}$, $j = 1, \ldots, J$ and associated weights $w_n^{(j)}$.

As can be seen from (4) and (5), obtaining the global estimate $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$ at each sensor presupposes that each sensor knows the JLF $f(\mathbf{z}_n|\mathbf{x}_n)$ as a function of the state $\mathbf{x}_n$. (Note that $\mathbf{z}_n$ is observed and thus fixed.) In particular, a PF approximation of $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$ relies on the pointwise evaluation of the JLF at the particles $\mathbf{x}_n^{(j)}$—i.e., on the evaluation of $f(\mathbf{z}_n|\mathbf{x}_n^{(j)})$—to obtain the weights $w_n^{(j)}$. Since each sensor knows only its local likelihood $f(\mathbf{z}_{n,k}|\mathbf{x}_n)$, we need a distributed method for calculating the JLF at each sensor. Such a method is proposed next.

III. Sequential Likelihood Consensus

The LC scheme introduced in [3], [5] performs a distributed approximate calculation of the JLF by means of an iterative consensus algorithm, using only communications between neighboring sensors. This scheme requires several consensus iterations per posterior update (5) (i.e., for each time n). In this section, we present the SLC as an evolution of the LC that requires only *one* consensus iteration per posterior update (5). This results in a substantial reduction of inter-sensor communications and latency. The SLC employs a novel dynamic consensus algorithm, whose recursion steps are interleaved with the posterior updates. This approach performs well if the time variation of the JLF is not too fast.

A. Approximation of the JLF

The SLC is based on an approximation of the JLF, which also underlies the LC [3]–[6]. Here, we will consider the generalized approximation proposed in [6], which does not assume a specific form of the local likelihood functions or of the local measurement functions. Assuming that $f(\mathbf{z}_{n,k}|\mathbf{x}_n) > 0$ for all $\mathbf{x}_n$, we can take the logarithm of (1), which yields

$$\log f(\mathbf{z}_n|\mathbf{x}_n) = \sum_{k=1}^{K} \log f(\mathbf{z}_{n,k}|\mathbf{x}_n)\,. \qquad (6)$$

A consensus-based distributed calculation of the sum in (6) is impossible because the terms of that sum depend on the unknown state $\mathbf{x}_n$. We therefore use the following basis expansion approximations of the local log-likelihood functions:

$$\log f(\mathbf{z}_{n,k}|\mathbf{x}_n) \approx \sum_{r=1}^{R} \alpha_{n,k,r}(\mathbf{z}_{n,k})\, \varphi_{n,r}(\mathbf{x}_n)\,, \qquad (7)$$

where the $\alpha_{n,k,r}(\mathbf{z}_{n,k})$ are expansion coefficients that contain all sensor-local information (in particular, the respective sensor measurement $\mathbf{z}_{n,k}$) and the $\varphi_{n,r}(\mathbf{x}_n)$ are fixed, sensor-independent basis functions that are assumed to be known to all sensors. A simple example of (7) would be a multivariate polynomial. Inserting (7) into (6), we obtain

$$\log f(\mathbf{z}_n|\mathbf{x}_n) \approx \sum_{r=1}^{R} a_{n,r}(\mathbf{z}_n)\, \varphi_{n,r}(\mathbf{x}_n)\,, \qquad (8)$$

with

$$a_{n,r}(\mathbf{z}_n) = \sum_{k=1}^{K} \alpha_{n,k,r}(\mathbf{z}_{n,k})\,. \qquad (9)$$

By exponentiating (8), locally at each sensor, we finally obtain the following approximation of the JLF, denoted $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$:

$$f(\mathbf{z}_n|\mathbf{x}_n) \approx \tilde{f}(\mathbf{z}_n|\mathbf{x}_n) \triangleq \exp\left(\sum_{r=1}^{R} a_{n,r}(\mathbf{z}_n)\, \varphi_{n,r}(\mathbf{x}_n)\right). \qquad (10)$$

Therefore, a sensor that knows the coefficients $a_{n,r}(\mathbf{z}_n)$ is able to evaluate the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ for all values of $\mathbf{x}_n$. Indeed, the collection of all $a_{n,r}(\mathbf{z}_n)$, i.e., $\mathbf{a}_n(\mathbf{z}_n) \triangleq (a_{n,1}(\mathbf{z}_n) \cdots a_{n,R}(\mathbf{z}_n))^\top$, can be viewed as a *sufficient statistic* [7] within the limits of the approximation (8).

Due to the sum expression (9), the sufficient statistic $\mathbf{a}_n(\mathbf{z}_n)$ can be computed in a distributed way using a consensus algorithm [10]. The resulting *LC scheme* for a distributed calculation of the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ requires each sensor to communicate only with its neighbors within a predetermined radius, and it avoids the use of complex routing protocols. However, the consensus algorithm must be sufficiently converged, which means that a sufficient number of consensus iterations must be executed for each posterior update (5) (i.e., for each n). Since each consensus iteration contributes to the latency and the amount of inter-sensor communications of the sequential state estimation scheme, this may be a limiting factor in practice, especially for real-time applications such as target tracking. Therefore, we next propose a novel dynamic consensus algorithm that allows an approximate sequential calculation of the sum in (9) using only a single consensus iteration for each n.

B. Dynamic Consensus Algorithm

Abstracting for a moment from our specific state estimation application, we consider a distributed tracking of a slowly time-varying sum $U_n = \sum_{k=1}^{K} u_{n,k}$, where $u_{n,k}$ is the input signal at sensor k. Let $s_{n,k}$ denote a time-dependent internal state at sensor k, initialized with $s_{0,k} = u_{0,k}$. At a given time $n \geq 1$,

a temporary internal state $\psi_{n,k}^{(0)}$ is first calculated at each sensor k from the previous state $s_{n-1,k}$ and the current and past inputs $\{u_{n-l,k}\}_{l=0}^{L-1}$ according to

$$\psi_{n,k}^{(0)} = \mu\, s_{n-1,k} + (1-\mu) \sum_{l=0}^{L-1} \omega_l\, u_{n-l,k}\,. \qquad (11)$$

Here, the ω_l are temporal weights (or filter coefficients) and $\mu \in [0,1]$ is a tuning parameter that controls the contribution of the previous internal state $s_{n-1,k}$ to $\psi_{n,k}^{(0)}$. Next, $I \geq 1$ consensus iterations are performed at each sensor, i.e.,

$$\psi_{n,k}^{(i)} = W_{k,k}\,\psi_{n,k}^{(i-1)} + \sum_{k' \in \mathcal{N}_k} W_{k,k'}\,\psi_{n,k'}^{(i-1)}\,, \quad i = 1, \ldots, I\,, \qquad (12)$$

where $\mathcal{N}_k$ denotes a predefined set of neighbors of sensor k and the $W_{k,l}$ are suitable weights that depend on the network topology [10]. The new internal state is given by $s_{n,k} = \psi_{n,k}^{(I)}$. Finally, the desired estimate of the sum U_n is obtained at sensor k as $\hat{U}_{n,k} = K s_{n,k}$. (The number of sensors K is assumed known to each sensor; distributed methods for computing K are available, e.g., [11].) We note that in the SLC, the number of iterations will be set to $I = 1$ (see Section III-C).

Using (11), (12), and $s_{n,k} = \psi_{n,k}^{(I)}$, the internal-state update relation for the entire network is obtained as

$$\mathbf{s}_n = \mathbf{W}^I \left[\mu\, \mathbf{s}_{n-1} + (1-\mu) \sum_{l=0}^{L-1} \omega_l\, \mathbf{u}_{n-l} \right], \qquad (13)$$

with the vectors $\mathbf{s}_n \triangleq (s_{n,1} \cdots s_{n,K})^\top$ and $\mathbf{u}_n \triangleq (u_{n,1} \cdots u_{n,K})^\top$ and the weight matrix $\mathbf{W} \triangleq (W_{k,l})$ [10]. This can be viewed as a generalization of the dynamic (or running) consensus algorithms proposed in [12]–[14]. The main difference is the fact that the input sequence $\mathbf{u}_n$ is filtered by a nonrecursive filter. Furthermore, the tuning parameter μ allows an adjustment of the influence of the previous internal state $\mathbf{s}_{n-1}$ relative to that of the filtered input, $\sum_{l=0}^{L-1} \omega_l\, \mathbf{u}_{n-l}$. Optimal procedures for choosing the filter length L, filter coefficients ω_l, tuning parameter μ, and number of consensus iterations I are an interesting subject for future research.

C. The SLC Algorithm

The SLC is an evolution of the LC in which $\mathbf{a}_n(\mathbf{z}_n)$ is updated sequentially, in a *single* iteration per time n, based on its previous value and the current measurements. The SLC uses the dynamic consensus algorithm of Section III-B with $I = 1$ to perform an approximate distributed calculation of the sum in (9). The SLC algorithm can be summarized as follows.

SEQUENTIAL LIKELIHOOD CONSENSUS (SLC) ALGORITHM

At each time step $n \geq 1$, each sensor k performs the following tasks:

1) The coefficients $\{\alpha_{n,k,r}(\mathbf{z}_{n,k})\}_{r=1}^{R}$ of the approximation (7) are calculated, e.g., by using least squares fitting as discussed in [4], [5].
2) *Dynamic consensus algorithm* — The following steps are executed for each $r \in \{1, \ldots, R\}$:
   a) If $n = 1$, the internal state of sensor k is initialized as $s_{0,k,r} = \alpha_{1,k,r}(\mathbf{z}_{1,k})$.
   b) A temporary internal state is computed according to (cf. (11))
   $$\psi_{n,k,r} = \mu\, s_{n-1,k,r} + (1-\mu)\, \alpha_{n,k,r}(\mathbf{z}_{n,k})\,, \qquad (14)$$
   where $\mu \in [0,1]$ is a tuning parameter.
   c) The temporary internal state $\psi_{n,k,r}$ is broadcast to all neighbors $k' \in \mathcal{N}_k$.
   d) The internal state $s_{n,k,r}$ is calculated from the local and neighboring temporary internal states, $\psi_{n,k,r}$ and $\{\psi_{n,k',r}\}_{k' \in \mathcal{N}_k}$, by a single consensus step (cf. (12)):
   $$s_{n,k,r} = W_{k,k}\,\psi_{n,k,r} + \sum_{k' \in \mathcal{N}_k} W_{k,k'}\,\psi_{n,k',r}\,. \qquad (15)$$
   The weights $W_{k,l}$ depend on the network topology [10]; here, we use Metropolis weights [15].
   e) The internal state $s_{n,k,r}$ is scaled as $\tilde{a}_{n,k,r}(\mathbf{z}_n) \triangleq K s_{n,k,r}$.
3) By substituting $\tilde{a}_{n,k,r}(\mathbf{z}_n)$ for $a_{n,r}(\mathbf{z}_n)$ and evaluating (10), sensor k is able to compute the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ for any value of $\mathbf{x}_n$. (We note that the $\tilde{a}_{n,k,r}(\mathbf{z}_n)$ for different k may differ slightly.)

Let $\mathbf{s}_{n,r} \triangleq (s_{n,1,r} \cdots s_{n,K,r})^\top$ collect the internal states of all sensors. Combining (14) and (15), the following internal-state update relation for the entire network is obtained:

$$\mathbf{s}_{n,r} = \mathbf{W} \left[\mu\, \mathbf{s}_{n-1,r} + (1-\mu)\, \boldsymbol{\alpha}_{n,r}(\mathbf{z}_n) \right], \qquad (16)$$

with $\boldsymbol{\alpha}_{n,r}(\mathbf{z}_n) \triangleq \big(\alpha_{n,1,r}(\mathbf{z}_{n,1}) \cdots \alpha_{n,K,r}(\mathbf{z}_{n,K})\big)^\top$. The SLC update (16) is a simplified form of the dynamic consensus algorithm (13), with input vector $\mathbf{u}_n \equiv \boldsymbol{\alpha}_{n,r}(\mathbf{z}_n)$, filter length $L = 1$, filter coefficient $\omega_0 = 1$, and a single consensus iteration, $I = 1$. As we will show in Section V, this simplified algorithm still leads to satisfactory performance if the time variation of $\boldsymbol{\alpha}_{n,r}(\mathbf{z}_n)$ is not too large.

The SLC scheme requires the transmission of R real numbers by each sensor at each time n. The (nonsequential) LC scheme [5] requires the transmission of RI real numbers by each sensor at each time n, where I is the number of consensus iterations performed within time step n. Therefore, using the SLC instead of the LC can lead to significant savings in inter-sensor communications, in addition to the reduced latency. The SLC and the LC share the advantage that the number of transmissions does not depend on the measurement dimensions $N_{n,k}$; this feature is particularly attractive in the case of high-dimensional measurements.

IV. SLC-BASED DISTRIBUTED PARTICLE FILTERING

We now show how the SLC can be used to obtain a distributed PF. As in our LC-based distributed PF [3], [5], each

sensor tracks a particle representation of the global posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ using a *local PF*. From the particle representation, a global state estimate $\hat{\mathbf{x}}_{n,k}$ approximating $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$ in (4) is computed. This estimate takes into account $\mathbf{z}_{1:n}$, i.e., the past and current measurements of *all* sensors. This requires each sensor to know the JLF $f(\mathbf{z}_n|\mathbf{x}_n)$ as a function of the state $\mathbf{x}_n$. Therefore, the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ in (10) is provided to each sensor in a distributed way by means of the SLC. The local PF algorithm at sensor k can be stated as follows.

LOCAL PARTICLE FILTER ALGORITHM

The local PF at sensor k is initialized at time $n=0$ by J particles $\mathbf{x}_{0,k}^{(j)}$, $j=1,\ldots,J$ drawn from the prior pdf $f(\mathbf{x}_0)$. At a given time $n\geq 1$, the following steps are performed:

1) For each previous particle $\mathbf{x}_{n-1,k}^{(j)}$, a new 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) The approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ is computed in a distributed way by means of the SLC as described in Section III. This step requires communications between neighboring sensors.
3) The weights associated with the particles $\bar{\mathbf{x}}_{n,k}^{(j)}$ drawn in Step 1 are calculated according to [9]
$$w_{n,k}^{(j)} = \frac{\tilde{f}(\mathbf{z}_n|\bar{\mathbf{x}}_{n,k}^{(j)})}{\sum_{j'=1}^{J}\tilde{f}(\mathbf{z}_n|\bar{\mathbf{x}}_{n,k}^{(j')})}, \quad j=1,\ldots,J.$$
This involves the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ computed in Step 2, which is evaluated at all particles $\bar{\mathbf{x}}_{n,k}^{(j)}$.
4) From the weighted particles $\{\bar{\mathbf{x}}_{n,k}^{(j)}, w_{n,k}^{(j)}\}_{j=1}^{J}$, an approximation of the MMSE state estimate (4) is computed as
$$\hat{\mathbf{x}}_{n,k} = \sum_{j=1}^{J} w_{n,k}^{(j)}\bar{\mathbf{x}}_{n,k}^{(j)}.$$
5) The set of particles and weights $\{\bar{\mathbf{x}}_{n,k}^{(j)}, w_{n,k}^{(j)}\}_{j=1}^{J}$ representing the current global posterior $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is *resampled* [9]; this produces J resampled particles $\mathbf{x}_{n,k}^{(j)}$ with identical weights. The $\mathbf{x}_{n,k}^{(j)}$ are obtained by sampling with replacement from the set $\{\bar{\mathbf{x}}_{n,k}^{(j')}\}_{j'=1}^{J}$, where $\bar{\mathbf{x}}_{n,k}^{(j')}$ is sampled with probability $w_{n,k}^{(j')}$.

This algorithm differs from the local PF algorithm of [3], [5] in that the SLC (rather than the LC) is used to provide the approximate JLF $\tilde{f}(\mathbf{z}_n|\mathbf{x}_n)$ to all sensors. This results in lower communication requirements and a smaller latency. Note that no communications between distant sensors and no complex routing protocols are required. Also, no particles, local state estimates, or measurements are communicated between the sensors. Because all local PF algorithms are identical, any differences between the state estimates $\hat{\mathbf{x}}_{n,k}$ at different sensors k are only due to the random sampling of the particles (using nonsynchronized local random generators) and errors introduced by the SLC method.

V. NUMERICAL STUDY

We will now assess the performance of the proposed SLC-based distributed PF for a target tracking problem.

A. Simulation Setup

The target is represented by a vector $\boldsymbol{\tau}_n = (x_n\ y_n\ \dot{x}_n\ \dot{y}_n)^\top$ containing the target's 2D position and 2D velocity in the x-y plane. The temporal evolution of $\boldsymbol{\tau}_n$ is given by $\boldsymbol{\tau}_n = \mathbf{G}\boldsymbol{\tau}_{n-1} + \mathbf{H}\mathbf{u}_n'$, $n=1,2,\ldots$, where
$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$
and the $\mathbf{u}_n' \in \mathbb{R}^2$ are independent and identically distributed (iid) according to $\mathbf{u}_n' \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{u'})$ with $\mathbf{C}_{u'} = \mathrm{diag}(10^{-10}, 10^{-10})$. However, assuming that the true target motion model is unknown to the sensors, we simulate the local PFs using the simple random walk state-space model $\mathbf{x}_n = \mathbf{x}_{n-1} + \mathbf{u}_n$, where the state $\mathbf{x}_n = (x_n\ y_n)^\top$ represents the position of the target and $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_u)$ with $\mathbf{C}_u = \mathrm{diag}(10^{-6}, 10^{-6})$.

The target emits a sound of constant amplitude $A=1$, which is sensed by $K=25$ acoustic amplitude sensors randomly deployed on a jittered grid within a rectangular region of (normalized) size 1×1. Each sensor communicates with other sensors within a radius of 0.45. The (scalar) measurement $z_{n,k}$ of sensor k is given by [16]
$$z_{n,k} = \frac{A}{\|\mathbf{x}_n - \boldsymbol{\xi}_{n,k}\|} + v_{n,k},$$
where $\boldsymbol{\xi}_{n,k}$ denotes the position of sensor k at time n and the $v_{n,k}$ are iid according to $v_{n,k} \sim \mathcal{N}(0, \sigma_v^2)$ with $\sigma_v^2 = 0.5$.

For the SLC, we approximate $\log f(\mathbf{z}_{n,k}|\mathbf{x}_n)$ by a multivariate polynomial of degree $R_p = 4$, which results in a basis expansion (7) of dimension $R = \binom{R_p+M}{R_p} - 1 = 14$. Therefore, 14 real numbers are broadcast by each sensor to its neighbors at each time n. The polynomial approximation is calculated by means of least squares fitting [4], [5]. For the consensus algorithm, we use Metropolis weights as specified in [15]. Each local PF employs $J=5000$ particles. We note that this number can be reduced by the scheme presented in [6].

The estimation performance is measured by the root-mean-square error of the state estimates $\hat{\mathbf{x}}_{n,k}$, denoted RMSE_n. At each time n, RMSE_n is computed as the square root of the averaged squared estimation error, where the averaging is over all sensors and 200 simulation runs. We also compute the *average* RMSE (ARMSE) by averaging RMSE_n^2 over all 1500 simulated time instants n and taking the square root.

B. Simulation Results

In Fig. 1, we show the dependence of the ARMSE of the proposed SLC-based distributed PF (abbreviated SLC-DPF) on the tuning parameter μ. It can be seen that the lowest ARMSE is achieved for $\mu = 0.7$. Therefore, unless stated otherwise, this value is used in the following.

Fig. 2 shows the temporal evolution of RMSE_n. The proposed SLC-DPF is compared with the LC-based distributed

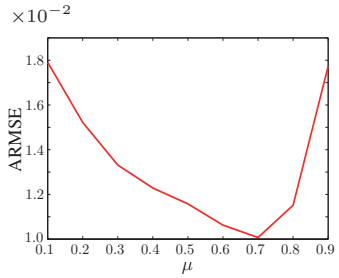

Figure 1. ARMSE of SLC-DPF versus tuning parameter μ.

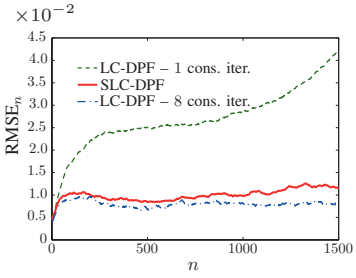

Figure 2. RMSE_n of SLC-DPF and LC-DPF versus time n.

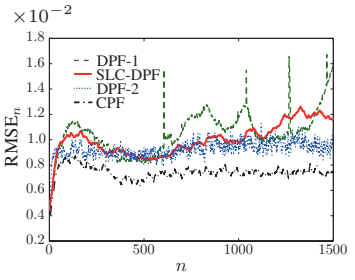

Figure 3. RMSE_n of SLC-DPF, state-of-the-art distributed PFs DPF-1 and DPF-2, and CPF versus time n.

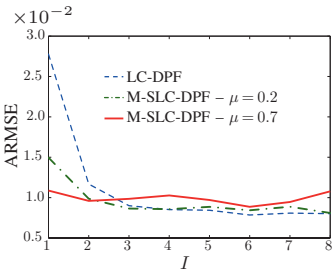

Figure 4. ARMSE of M-SLC-DPF and LC-DPF versus number of consensus iterations I.

PF of [5] (abbreviated LC-DPF), which uses one or eight consensus iterations. SLC-DPF performs significantly better than LC-DPF with one consensus iteration; note that the communication requirements are equal in this case. Furthermore, SLC-DPF performs slightly worse than LC-DPF with eight iterations; however, here the communication requirements and latency of SLC-DPF are lower by a factor of eight.

In Fig. 3, we compare SLC-DPF with the state-of-the-art distributed PFs of [17] and [18] (referred to as DPF-1 and DPF-2, respectively) and with a centralized PF (CPF) that processes all measurements at a fusion center. All PFs use $J = 5000$ particles. Both DPF-1 and DPF-2 use eight consensus iterations. The performance of SLC-DPF is not dramatically poorer than that of CPF, and roughly similar to that of DPF-1 and DPF-2. However, the communication requirements of SLC-DPF are significantly lower than those of DPF-1 and DPF-2: the total counts of real numbers transmitted per time step by SLC-DPF, DPF-1, and DPF-2 in the entire network (all sensors) are 350, 10^3, and 10^6, respectively.

Finally, we consider a modified SLC-DPF (abbreviated M-SLC-DPF) that employs multiple consensus iterations per time step n. That is, $I \geq 1$, similar to LC-DPF and, also, to the general dynamic consensus in (13). Note that LC-DPF is a special case of M-SLC-DPF with $\mu = 0$. In Fig. 4, we compare the ARMSE of M-SLC-DPF (with $\mu = 0.7$ and $\mu = 0.2$) and LC-DPF, plotted as a function of the number I of consensus iterations. For both methods, each sensor broadcasts RI real numbers to its neighbors. It is seen that for $I = 1$ and $I = 2$, M-SLC-DPF with $\mu = 0.7$ performs best; for $I = 3$ and $I = 4$, M-SLC-DPF with $\mu = 0.2$ performs best; and for $I \geq 5$, LC-DPF (i.e., $\mu = 0$) performs best. These results suggest that the optimal value of μ decreases to 0 as I increases.

VI. Conclusion

We proposed a distributed method for the sequential computation of the joint (all-sensors) likelihood function (JLF) in a wireless sensor network. This *sequential likelihood consensus* (SLC) method uses a novel dynamic consensus algorithm to update, at each sensor, an approximate parametric description of the JLF. We furthermore proposed the use of the SLC method in a reduced-communications, reduced-latency scheme for distributed particle filtering. At each sensor, a local particle filter computes a global state estimate. Based on the JLF provided by the SLC, this estimate takes into account the measurements of all sensors. Simulation results obtained for a target tracking problem demonstrated good performance of the proposed distributed particle filter.

References

[1] H. J. Rad, B. Abolhassani, and M. Abdizadeh, "An energy efficient target tracking scheme for distributed wireless sensor networks," in *Proc. ISWCS-09*, (Siena, Italy), pp. 136–140, Sep. 2009.

[2] G. Ferrari, *Sensor Networks: Where Theory Meets Practice*. Heidelberg, Germany: Springer, 2010.

[3] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus: Principles and application to distributed particle filtering," in *Proc. 44th Asilomar Conf. Sig., Syst., Comp.*, (Pacific Grove, CA), pp. 349–353, Nov. 2010.

[4] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Distributed Gaussian particle filtering using likelihood consensus," in *Proc. IEEE ICASSP-11*, (Prague, Czech Republic), pp. 3756–3759, May 2011.

[5] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djurić, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering." Available online: arXiv:1108.6214v1 [stat.AP], Aug. 2011.

[6] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Likelihood consensus-based distributed particle filtering with distributed proposal density adaptation." Available online: arXiv:1109.6191v1 [stat.AP], Sep. 2011.

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

[8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA: Artech House, 2004.

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

[10] 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.

[11] D. Mosk-Aoyama and D. Shah, "Fast distributed algorithms for computing separable functions," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2997–3007, Jul. 2008.

[12] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congress*, (Prague, Czech Republic), Jul. 2005.

[13] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[14] P. Braca, S. Marano, and V. Matta, "Running consensus in wireless sensor networks," in *Proc. Int. Conf. Inf. Fusion (FUSION 2008)*, (Cologne, Germany), pp. 152–157, Jun.–Jul. 2008.

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

[16] 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.

[17] A. Mohammadi and A. Asif, "Consensus-based distributed unscented particle filter," in *Proc. IEEE SSP-11*, (Nice, France), pp. 237–240, Jun. 2011.

[18] 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.