TARGET TRACKING USING A DISTRIBUTED PARTICLE-PDA FILTER WITH SPARSITY-PROMOTING LIKELIHOOD CONSENSUS

Rene Repp[⋆]*, Pavel Rajmic*[†]*, Florian Meyer*[‡]*, and Franz Hlawatsch*[⋆]

[⋆]Institute of Telecommunications, TU Wien, Vienna, Austria
[†]Department of Telecommunications, Brno University of Technology, Brno, Czech Republic
[‡]LIDS, Massachusetts Institute of Technology, Cambridge, MA, USA

ABSTRACT

We propose a distributed particle-based probabilistic data association filter (PDAF) for target tracking in the presence of clutter and missed detections. The proposed PDAF employs a new "sparsity-promoting" likelihood consensus that uses the orthogonal matching pursuit for a sparse approximation of the local likelihood functions. Simulation results demonstrate that, compared to the conventional likelihood consensus based on least-squares approximation, large savings in intersensor communication can be obtained without compromising the tracking performance.

Index Terms— Distributed target tracking, sensor network, probabilistic data association, likelihood consensus, orthogonal matching pursuit.

1. INTRODUCTION

Tracking a moving target under measurement-origin uncertainty [1–3] is an important problem in many applications [4, 5]. A popular Bayesian approach to this problem is the probabilistic data association filter (PDAF) [4]. Here, we consider a generalized PDAF that uses a particle implementation to accommodate nonlinear and non-Gaussian state-space models [6, 7], and we furthermore consider a distributed multisensor mode of operation within a decentralized sensor network. Distributed multisensor filters have numerous advantages related to performance, robustness, computational complexity, scalability, and communication cost [8].

We propose a distributed particle-based multisensor PDAF that relies on the distributed particle filtering algorithm presented in [9, 10] but uses a new "sparsity-promoting" variant of the likelihood consensus (LC) to reduce intersensor communication. The proposed distributed PDAF extends the distributed particle filter of [9, 10] to scenarios with clutter and missed detections. The proposed sparsity-promoting LC uses the orthogonal matching pursuit (OMP) [11, 12] for local likelihood function approximation. The OMP allows an easy specification of the number of significant approximation coefficients, which leads to a flexible tradeoff between approximation accuracy, computational complexity, and communication cost. Our simulation results demonstrate that using the OMP, large savings in intersensor communication can be obtained without compromising the tracking performance.

This paper is organized as follows. The system model is described in Section 2. In Section 3, we develop the distributed particle-based PDAF, and in Section 4, we present the sparsity-promoting LC scheme. Finally, simulation results are reported in Section 5.

This work was supported in part by the Austrian Science Fund (FWF) under grants P27370-N30 and J3886-N31, by the Czech Science Foundation (GAČR) under grant 17-19638S, and by the Czech National Sustainability Program under grant LO1401. For the research, the infrastructure of the SIX Center was used.

2. SYSTEM MODEL AND STATISTICAL FORMULATION

We consider an object ("target") with an unknown time-varying state $\mathbf{x}_n \in \mathbb{R}^d$, where $n \in \mathbb{N}_0$ is a discrete time index. The state evolves according to a Markovian dynamic model with a known state-transition probability density function (pdf) $f(\mathbf{x}_n|\mathbf{x}_{n-1})$. There are S possibly heterogeneous sensors, and each sensor $s \in \{1,\dots,S\}$ is able to communicate with a certain set of "neighbor" sensors, $\mathcal{N}_s \subseteq \{1,\dots,S\} \setminus \{s\}$. We assume that the corresponding communication graph is connected, i.e., there is a connection—possibly composed of multiple hops—between any two sensors.

Each sensor suffers from a measurement-origin uncertainty that results from clutter and missed detections. Specifically, at time n, sensor $s \in \{1,\dots,S\}$ produces a random number $M_n^{(s)}$ of measurements $\mathbf{z}_{n,m}^{(s)}$, $m \in \{1,\dots,M_n^{(s)}\}$ via a detection process. These measurements consist of at most one target-originated measurement and unwanted clutter measurements (false alarms). For the derivation of the local likelihood function, we make the following assumptions about the measurement model [1, 2]: (A1) At time n, the target is detected by sensor s with probability $P_{\mathrm{d}}^{(s)}(\mathbf{x}_n)$. If detected by sensor s, the target generates exactly one measurement $\mathbf{z}_{n,m}^{(s)}$ at sensor s according to the conditional pdf $f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,m}^{(s)}\big|\mathbf{x}_n\big)$. (A2) For each sensor s, the number of clutter measurements is Poisson distributed with mean $\mu^{(s)}$, and the clutter measurements are independent and identically distributed (iid) with pdf $f_{\mathrm{c}}^{(s)}(\cdot)$. (A3) The sensors do not know whether their measurements are target-originated or clutter. (A4) The measurements of different sensors, $\mathbf{z}_n^{(s)} \triangleq \big(\mathbf{z}_{n,1}^{(s)},\dots,\mathbf{z}_{n,M_n^{(s)}}^{(s)}\big)$, are conditionally independent given the state $\mathbf{x}_n$. With these assumptions, the local likelihood function (LLF) of sensor s is given by [2, Sec. 4.5]

$$f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) = C\big(\mathbf{z}_n^{(s)}\big)\Bigg[\big(1-P_{\mathrm{d}}^{(s)}(\mathbf{x}_n)\big)\,\mu^{(s)} + P_{\mathrm{d}}^{(s)}(\mathbf{x}_n)\sum_{m=1}^{M_n^{(s)}}\frac{f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,m}^{(s)}\big|\mathbf{x}_n\big)}{f_{\mathrm{c}}^{(s)}\big(\mathbf{z}_{n,m}^{(s)}\big)}\Bigg], \tag{1}$$

where $C\big(\mathbf{z}_n^{(s)}\big)$ is a normalization constant. Without clutter and missed detections (i.e., $M_n^{(s)}=1$, $\mu^{(s)}=0$, $P_{\mathrm{d}}^{(s)}(\mathbf{x}_n)=1$), the LLF in (1) would simplify to $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) = f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$.

For optimum Bayesian estimation of the state $\mathbf{x}_n$ based on the measurements of all sensors, the *global likelihood function* (GLF) $f(\mathbf{z}_n|\mathbf{x}_n)$, with $\mathbf{z}_n \triangleq \big(\mathbf{z}_n^{(1)},\dots,\mathbf{z}_n^{(S)}\big)$, is needed. Due to assumption (A4), the GLF is obtained as

$$f(\mathbf{z}_n|\mathbf{x}_n) = \prod_{s=1}^{S} f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big). \tag{2}$$

3. DISTRIBUTED PARTICLE-PDAF

At time n, we would like to estimate the time-varying target state $\mathbf{x}_n$ from the measurements of all the sensors up to time n, $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1, \ldots, \mathbf{z}_n)$. In an optimum Bayesian framework, this can be done by the minimum mean-square error (MMSE) estimator [13]

$$\hat{\mathbf{x}}_n^{\mathrm{MMSE}} \triangleq \mathrm{E}\{\mathbf{x}_n|\mathbf{z}_{1:n}\} = \int \mathbf{x}_n \, f(\mathbf{x}_n|\mathbf{z}_{1:n}) \, \mathrm{d}\mathbf{x}_n \,. \tag{3}$$

Here, $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is the *global posterior pdf*. Our main task is now to compute $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ in a sequential (recursive) fashion. This is accomplished by a generalization of the PDAF [4] to arbitrary—i.e., generally non-Gaussian—pdfs [2]. For a practical implementation of this generalized PDAF, all pdfs are represented by particles and weights. In a distributed version of this particle-based PDAF, each sensor s runs a *local particle filter* (LPF). This LPF uses a local approximation of the GLF, $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n) \approx f(\mathbf{z}_n|\mathbf{x}_n)$, and not merely its own LLF $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$. Thereby, the resulting local state estimate $\hat{\mathbf{x}}_n^{(s)}$ takes into account the measurements of all the sensors, $\mathbf{z}_n$. The LC-based distributed calculation of $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ will be discussed in Section 4.

In the LPF at sensor s, the global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is represented by a set of particles and associated weights $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^J$, with $\sum_{j=1}^J w_n^{(s,j)} = 1$. Accordingly, the recursive update of the global posterior pdf, $f(\mathbf{x}_{n-1}|\mathbf{z}_{1:n-1}) \to f(\mathbf{x}_n|\mathbf{z}_{1:n})$, amounts to a recursive update of the associated "weighted particle set," $\big\{\big(\mathbf{x}_{n-1}^{(s,j)}, w_{n-1}^{(s,j)}\big)\big\}_{j=1}^J \to \big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^J$. We here consider only the simplest particle filter algorithm, the so-called sequential importance resampling filter [7]. At time $n \geq 1$, the following steps are performed. First, for each previous particle $\mathbf{x}_{n-1}^{(s,j)}$, a new particle $\mathbf{x}_n^{(s,j)}$ is sampled from $f\big(\mathbf{x}_n\big|\mathbf{x}_{n-1}^{(s,j)}\big)$, i.e., from the state-transition pdf $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ evaluated at $\mathbf{x}_{n-1} = \mathbf{x}_{n-1}^{(s,j)}$. Next, the weights associated with the particles $\mathbf{x}_n^{(s,j)}$ are calculated as

$$w_n^{(s,j)} = c\, \hat{f}_s\big(\mathbf{z}_n\big|\mathbf{x}_n^{(s,j)}\big)\,, \quad j = 1, \ldots, J, \tag{4}$$

with $c = 1/\sum_{j=1}^J \hat{f}_s\big(\mathbf{z}_n\big|\mathbf{x}_n^{(s,j)}\big)$. Note that this expression involves the local GLF approximation $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ and, thereby, the current measurements of all the sensors, $\mathbf{z}_n$. Furthermore, if a suitable criterion is satisfied (as discussed in [7, 14]), the set $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^J$ is resampled to avoid an effect known as particle degeneracy. From the final weighted particle set $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^J$, a Monte Carlo approximation of the global MMSE state estimate (3) is now computed as

$$\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^J w_n^{(s,j)} \, \mathbf{x}_n^{(s,j)}.$$

This recursive algorithm is initialized at time $n = 0$ by randomly drawing J particles $\mathbf{x}_0^{(s,j)}$ from a prior pdf $f(\mathbf{x}_0)$, and the weights are set to $w_0^{(s,j)} \equiv 1/J$.

Through the above LPF recursion, at each time n, each sensor s obtains an approximate particle representation $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^J$ of the global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ and an approximation $\hat{\mathbf{x}}_n^{(s)}$ of the MMSE state estimate $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$; both involve the measurements of all the sensors. The LPF algorithms running at different sensors are identical. Therefore, any differences between the state estimates $\hat{\mathbf{x}}_n^{(s)}$ at different sensors s are only due to the random sampling of the particles at each sensor and errors caused by the LC (e.g., because of incompletely converged consensus algorithms).

4. SPARSITY-PROMOTING LIKELIHOOD CONSENSUS

We now discuss the distributed calculation of the GLF approximations $\hat{f}_s\big(\mathbf{z}_n\big|\mathbf{x}_n^{(s,j)}\big)$ required in the weight calculation step (4).

4.1. Review of the Likelihood Consensus Scheme

We can formally write the GLF $f(\mathbf{z}_n|\mathbf{x}_n)$ as

$$f(\mathbf{z}_n|\mathbf{x}_n) = \exp\big(\ln f(\mathbf{z}_n|\mathbf{x}_n)\big) = \exp\big(S\,\lambda(\mathbf{x}_n)\big)\,, \tag{5}$$

with $\lambda(\mathbf{x}_n) \triangleq \ln f(\mathbf{z}_n|\mathbf{x}_n)/S$. Using (2), we obtain

$$\lambda(\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^S \ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)\,. \tag{6}$$

This is the average of the *log-LLFs* $\ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ of the individual sensors $s = 1, \ldots, S$, which are functions of $\mathbf{x}_n \in \mathbb{R}^d$ (note that $\mathbf{z}_n^{(s)}$ is observed and thus considered fixed).

In the LC scheme [10], for a distributed calculation of $\lambda(\mathbf{x}_n)$, each log-LLF is approximated by a finite-order function expansion, i.e.,

$$\ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) \approx \sum_{k=1}^K \alpha_k^{(s)}\, \psi_k(\mathbf{x}_n)\,, \tag{7}$$

with fixed functions ("atoms") $\psi_k(\cdot)$ that are identical at all sensors and known to all sensors. A specific choice of the "dictionary" $\big\{\psi_k(\cdot)\big\}_{k=1}^K$ will be considered in Section 5. The calculation of the expansion coefficients $\big\{\alpha_k^{(s)}\big\}_{k=1}^K$ will be discussed in Section 4.2; note that the $\alpha_k^{(s)}$ depend on the local measurement $\mathbf{z}_n^{(s)}$. Inserting (7) into (6) and changing the order of summations yields

$$\lambda(\mathbf{x}_n) \approx \frac{1}{S}\sum_{s=1}^S\sum_{k=1}^K \alpha_k^{(s)}\, \psi_k(\mathbf{x}_n) = \sum_{k=1}^K \beta_k\, \psi_k(\mathbf{x}_n)\,, \tag{8}$$

with the global expansion coefficients

$$\beta_k \triangleq \frac{1}{S}\sum_{s=1}^S \alpha_k^{(s)}\,, \quad k = 1, \ldots, K. \tag{9}$$

By (8), distributed calculation of $\lambda(\mathbf{x}_n)$ amounts to distributed calculation of the global coefficients β_k for $k = 1, \ldots, K$. Based on expression (9), this can be done by executing K instances of the *average consensus algorithm* [15], using only communication with the neighbor sensors $s' \in \mathcal{N}_s$. In iteration $i \in \{1, 2, \ldots\}$ of the kth instance of the average consensus algorithm (which is used to calculate β_k), sensor s updates an iterated estimate $\hat{\beta}_{k,i}^{(s)}$ of β_k as

$$\hat{\beta}_{k,i}^{(s)} = \sum_{s' \in \{s\}\cup\mathcal{N}_s} \gamma_{s,s'}\, \hat{\beta}_{k,i-1}^{(s')}\,, \tag{10}$$

where the $\gamma_{s,s'}$ are suitably chosen weights [15–17]. The recursion (10) is initialized by the local expansion coefficient, i.e., $\hat{\beta}_{k,0}^{(s)} = \alpha_k^{(s)}$. Note that (10) implies that in each iteration i, sensor s has to broadcast its iterated estimates $\hat{\beta}_{k,i}^{(s)}$, $k = 1, \ldots, K$ to its neighbors $s' \in \mathcal{N}_s$. As an alternative to the average consensus algorithm, K instances of a gossip algorithm [18] can be used.

If the communication graph is connected (as we assumed in Section 2), then for $i \to \infty$ the consensus recursion (10) is guaranteed to converge to β_k [16]. In practice, only a finite number I of iterations is executed, where I is fixed or chosen adaptively based on some

stopping criterion. The resulting final estimate $\hat{\beta}_{k,I}^{(s)}$ then provides an approximation of β_k. Thereby, we obtain a corresponding approximation of $\lambda(\mathbf{x}_n)$ in (8), $\lambda(\mathbf{x}_n) \approx \hat{\lambda}_s(\mathbf{x}_n) \triangleq \sum_{k=1}^{K} \hat{\beta}_{k,I}^{(s)} \psi_k(\mathbf{x}_n)$, which entails the desired GLF approximation $\hat{f}_s\big(\mathbf{z}_n|\mathbf{x}_n^{(s,j)}\big)$ via (5):

$$\begin{aligned}\hat{f}_s\big(\mathbf{z}_n|\mathbf{x}_n^{(s,j)}\big) &= \exp\big(S\hat{\lambda}_s\big(\mathbf{x}_n^{(s,j)}\big)\big) \\ &= \exp\left(S\sum_{k=1}^{K}\hat{\beta}_{k,I}^{(s)}\psi_k\big(\mathbf{x}_n^{(s,j)}\big)\right).\end{aligned}$$

In the course of the I consensus iterations, each sensor has to broadcast IK real numbers to its neighbors. This communication cost can be reduced by an adaptive pruning of the local expansion coefficients $\alpha_k^{(s)}$, as explained in the next subsection.

4.2. OMP-based LLF Approximation

The standard way to calculate the local expansion coefficients $\alpha_k^{(s)}$, $k=1,\ldots,K$ in (7) at the respective sensor s is via a least-squares (LS) fit of $\sum_{k=1}^{K}\alpha_k^{(s)}\psi_k(\mathbf{x}_n)$ to the log-LLF $\ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ [9, 10]. Because only the values of the log-LLF at the current particles $\mathbf{x}_n^{(s,j)}$ are needed in (4), the approximation of the log-LLF has to be good only in the state-space regions containing the $\mathbf{x}_n^{(s,j)}$. Therefore, the LS fit minimizes, with respect to the the local coefficient vector $\boldsymbol{\alpha}^{(s)} \triangleq \big(\alpha_1^{(s)} \cdots \alpha_K^{(s)}\big)^{\mathrm{T}}$, the approximation error of the expansion (7) evaluated at the particles $\mathbf{x}_n^{(s,j)}$, $j=1,\ldots,J$, i.e., $\big\|\boldsymbol{\lambda}^{(s)} - \sum_{k=1}^{K}\alpha_k^{(s)}\boldsymbol{\psi}_k^{(s)}\big\|$, where $\boldsymbol{\lambda}^{(s)} \triangleq \big(\ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n^{(s,1)}\big) \cdots \ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n^{(s,J)}\big)\big)^{\mathrm{T}}$ and $\boldsymbol{\psi}_k^{(s)} \triangleq \big(\psi_k\big(\mathbf{x}_n^{(s,1)}\big) \cdots \psi_k\big(\mathbf{x}_n^{(s,J)}\big)\big)^{\mathrm{T}}$. This error can also be written as $\big\|\boldsymbol{\lambda}^{(s)} - \boldsymbol{\Psi}^{(s)}\boldsymbol{\alpha}^{(s)}\big\|$, where $\boldsymbol{\Psi}^{(s)}$ is the $J\times K$ matrix with columns $\boldsymbol{\psi}_1^{(s)},\ldots,\boldsymbol{\psi}_K^{(s)}$. Assuming that $J \geq K$ (i.e., there are at least as many particles as expansion coefficients) and that the $\boldsymbol{\psi}_k^{(s)}$ are linearly independent, the solution to this minimization problem is given by $\boldsymbol{\alpha}_{\mathrm{LS}}^{(s)} = \boldsymbol{\Psi}^{(s)\#}\boldsymbol{\lambda}^{(s)}$ with $\boldsymbol{\Psi}^{(s)\#} \triangleq \big(\boldsymbol{\Psi}^{(s)\mathrm{T}}\boldsymbol{\Psi}^{(s)}\big)^{-1}\boldsymbol{\Psi}^{(s)\mathrm{T}}$ [19].

We now propose a sparse alternative to the LS fit that uses an iterative algorithm known as the orthogonal matching pursuit (OMP) [11, 12]. The idea is to reduce the number of "significant" expansion coefficients and, thereby, the number of consensus instances and, in turn, the communication cost of the LC. In the first iteration, the OMP selects the atom $\psi_{k_1}(\cdot)$ that best matches $\ln f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ evaluated at the particles $\mathbf{x}_n^{(s,j)}$, i.e.,

$$k_1 = \operatorname*{argmax}_{k\in\{1,\ldots,K\}} \frac{\big|\boldsymbol{\psi}_k^{(s)\mathrm{T}}\boldsymbol{\lambda}^{(s)}\big|}{\|\boldsymbol{\psi}_k^{(s)}\|}.$$

Then, a residual $\boldsymbol{\rho}_1$ is formed by subtracting from $\boldsymbol{\lambda}^{(s)}$ the orthogonal projection of $\boldsymbol{\lambda}^{(s)}$ onto $\boldsymbol{\psi}_{k_1}^{(s)}$, i.e., $\boldsymbol{\rho}_1 = \boldsymbol{\lambda}^{(s)} - \big(\boldsymbol{\psi}_{k_1}^{(s)\mathrm{T}}\boldsymbol{\lambda}^{(s)} / \|\boldsymbol{\psi}_{k_1}^{(s)}\|\big)\boldsymbol{\psi}_{k_1}^{(s)}$. In the second iteration, the atom $\psi_{k_2}(\cdot)$ that best matches the residual $\boldsymbol{\rho}_1$ is selected, and a new residual $\boldsymbol{\rho}_2$ is formed by subtracting from $\boldsymbol{\lambda}^{(s)}$ the orthogonal projection of $\boldsymbol{\lambda}^{(s)}$ onto the linear subspace of $\mathbb{R}^J$ spanned by $\boldsymbol{\psi}_{k_1}^{(s)}$ and $\boldsymbol{\psi}_{k_2}^{(s)}$. More generally, in iteration $\ell \geq 2$, the atom $\psi_{k_\ell}(\cdot)$ that best matches the residual $\boldsymbol{\rho}_{\ell-1}$ is selected, i.e.,

$$k_\ell = \operatorname*{argmax}_{k\in\mathcal{K}_\ell} \frac{\big|\boldsymbol{\psi}_k^{(s)\mathrm{T}}\boldsymbol{\rho}_{\ell-1}\big|}{\|\boldsymbol{\psi}_k^{(s)}\|},$$

with $\mathcal{K}_\ell \triangleq \{1,\ldots,K\} \setminus \{k_1,\ldots,k_{\ell-1}\}$. Then, the new residual $\boldsymbol{\rho}_\ell$ is formed as $\boldsymbol{\rho}_\ell = \boldsymbol{\lambda}^{(s)} - \mathbf{P}_\ell\boldsymbol{\lambda}^{(s)}$, with the orthogonal projection matrix $\mathbf{P}_\ell \triangleq \boldsymbol{\Psi}_\ell^{(s)}\boldsymbol{\Psi}_\ell^{(s)\#}$, where $\boldsymbol{\Psi}_\ell^{(s)}$ is the $J\times(\ell-1)$ matrix with columns $\boldsymbol{\psi}_{k_1}^{(s)},\ldots,\boldsymbol{\psi}_{k_{\ell-1}}^{(s)}$. Note that $\boldsymbol{\rho}_\ell$ is orthogonal to all the previously selected atom vectors $\boldsymbol{\psi}_{k_1}^{(s)},\ldots,\boldsymbol{\psi}_{k_{\ell-1}}^{(s)}$. The residual can also be written (and calculated) as $\boldsymbol{\rho}_\ell = \boldsymbol{\lambda}^{(s)} - \boldsymbol{\Psi}_\ell^{(s)}\mathbf{c}_\ell$, where $\mathbf{c}_\ell \triangleq \boldsymbol{\Psi}_\ell^{(s)\#}\boldsymbol{\lambda}^{(s)}$ corresponds to the LS fit of $\boldsymbol{\Psi}_\ell^{(s)}\mathbf{c}$ to $\boldsymbol{\lambda}^{(s)}$. We note that in contrast to the LS-based calculation of $\boldsymbol{\alpha}^{(s)}$, there may be $J<K$ (i.e., fewer particles than expansion coefficients).

This iterative algorithm is stopped after $L \leq K$ iterations, where L is either predetermined or chosen adaptively by the criterion that $\|\boldsymbol{\rho}_\ell\|$ becomes smaller than some threshold. The result is the coefficient vector $\boldsymbol{\alpha}_{\mathrm{OMP}}^{(s)} = \big(\alpha_{\mathrm{OMP},1}^{(s)} \cdots \alpha_{\mathrm{OMP},K}^{(s)}\big)^{\mathrm{T}}$ whose elements $\alpha_{\mathrm{OMP},k}^{(s)}$ are $\alpha_{\mathrm{OMP},k_\ell}^{(s)} = (\mathbf{c}_L)_\ell$ for $\ell=1,\ldots,L$ and zero otherwise. Thus, only L elements of $\boldsymbol{\alpha}_{\mathrm{OMP}}^{(s)}$ are nonzero. As a consequence, in the first consensus iteration in the LC, each sensor broadcasts only $L \leq K$ nonzero coefficients (along with their indices k_ℓ) to its neighbors. In the consensus update (10), all coefficients that were not transmitted by a neighbor are treated as zero. In the course of progressing consensus iterations, since the sets of the indices k_ℓ corresponding to nonzero coefficients at different sensors are not exactly equal in general, the number of nonzero coefficients $\hat{\beta}_{k,i}^{(s)}$ to be broadcast will increase somewhat beyond L. However, as verified experimentally in Section 5, this number tends to be still well below K.

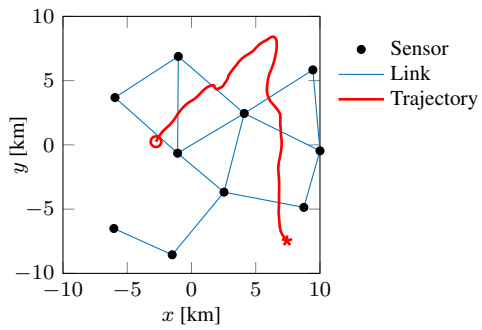

Fig. 1. Surveillance region with sensor network and target trajectory.

5. SIMULATION RESULTS

We simulated a network consisting of ten sensors located in a 2-D surveillance region of dimension 20 km × 20 km. The state vector is $\mathbf{x}_n \triangleq (x_n\ y_n\ \dot{x}_n\ \dot{y}_n)^{\mathrm{T}}$, where x_n and y_n are the positions and $\dot{x}_n$ and $\dot{y}_n$ the velocities of the target in the two coordinate directions. Fig. 1 shows the surveillance region, the sensor network, and the target trajectory used for all simulation runs. The state vector is assumed to evolve according to the near-constant-velocity model $\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{G}\mathbf{u}_n$ [20, Sec. 6.3.2]. Here, the 4×4 matrix $\mathbf{F}$ and the 4×2 matrix $\mathbf{G}$ are defined as in [20, Sec. 6.3.2] (the time scan duration involved in these matrices is chosen as $T=40$ s), and the 2-D driving process $\mathbf{u}_n$ is modeled as an iid zero-mean Gaussian random process with variance $\sigma_u^2 = 10^{-3}\,\mathrm{m}^2/\mathrm{s}^4$. The initial prior pdf $f(\mathbf{x}_0)$ is an uncorrelated Gaussian pdf with mean $(7.5\,\mathrm{km}\ \ -7.5\,\mathrm{km}\ \ -2\,\mathrm{m/s}\ \ 2\,\mathrm{m/s})^{\mathrm{T}}$, position standard deviation 100 m, and velocity standard deviation 2 m/s.

The sensors produce range and bearing measurements with detection probability $P_{\mathrm{d}}^{(s)}(\mathbf{x}_n) = 0.9$. The target-originated measurements are corrupted by zero-mean Gaussian noise with range standard deviation 150 m and bearing standard deviation 1°. The clutter measurements are uniformly distributed in the surveillance region with mean number of clutter measurements $\mu^{(s)} = 5$. We performed 1000 simulation runs using the target trajectory shown in Fig. 1 and different realizations of measurements and particles.

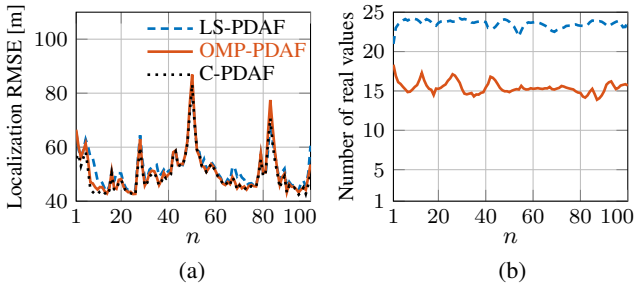

Fig. 2. Simulation results for $L=10$: (a) Localization RMSE (without lost tracks) versus time, (b) average number of real values broadcast per sensor and consensus iteration versus time.

The proposed distributed PDAF (abbreviated OMP-PDAF) calculates L expansion coefficients at each sensor by means of L OMP iterations. For comparison, we also simulated a distributed PDAF that uses LS fits to calculate L significant expansion coefficients (abbreviated LS-PDAF). More specifically, LS-PDAF first computes a full LS fit but keeps only the L coefficients with the highest absolute values, and then calculates a new LS fit using only the corresponding L atoms. In both filters, the LC uses a 2-D Fourier dictionary with $K=25$ atoms given by [10] $\psi_k(\mathbf{x}_n)=\tilde{\psi}_{\tilde{k}_1}(x_n)\tilde{\psi}_{\tilde{k}_2}(y_n)$, with an index transformation that maps $k\in\{1,\ldots,25\}$ to $(\tilde{k}_1,\tilde{k}_2)\in\{1,\ldots,5\}^2$. Here, the 1-D atoms $\tilde{\psi}_{\tilde{k}}(x)$ are given by

$$\tilde{\psi}_{\tilde{k}}(x)=\begin{cases}1, & \tilde{k}=1,\\ \cos\left(\frac{2\pi}{d_a}(\tilde{k}-1)x\right), & \tilde{k}=2,3,\\ \sin\left(\frac{2\pi}{d_a}(\tilde{k}-3)x\right), & \tilde{k}=4,5,\end{cases}$$

where $d_a=20\,\text{km}$ is the width of the surveillance area in each coordinate direction. (Note that we use a 2-D, rather than 4-D, dictionary because the sensors produce range and bearing measurements and thus the LLFs depend only on the x_n and y_n components of the state.) Furthermore, the LC uses Metropolis weights $\gamma_{s,s'}$ [9,10] and performs $I=100$ consensus iterations. As a performance benchmark, we also simulated a centralized multisensor PDAF, abbreviated C-PDAF, which has access to all the sensor measurements and thus uses in (4) the exact GLF $f(\mathbf{z}_n|\mathbf{x}_n)$ given by (2) and (1).

Fig. 2(a) shows the localization root-mean-square error (RMSE) versus time for $L=10$. In the calculation of the RMSE, simulation runs where the target track was lost were omitted; such a track loss was declared if the localization error norm exceeded 1 km at least at one time point n. (The percentage of lost tracks will be shown separately later.) It can be seen that both OMP-PDAF and LS-PDAF perform similar to C-PDAF, with OMP-PDAF performing slightly better than LS-PDAF in certain time intervals. The peaks in the localization RMSE around times $n=28$, 50, and 83 can be attributed to the three target maneuvers visible in Fig. 1.

The average number of real values broadcast by one sensor during one consensus iteration is a measure of the amount of communication required. For $L=10$, Fig. 2(b) shows this average number of real values, averaged over all consensus iterations and sensors, versus time. At most times, only about 15 real values are broadcast in OMP-PDAF whereas about 24—i.e., almost all of the 25 available coefficients—are broadcast in LS-PDAF. Thus, the use of the OMP leads to a reduction of communication by about 38%.

Fig. 3(a) shows the time-averaged localization RMSE, again without lost tracks, versus L. The RMSEs for OMP-PDAF and LS-PDAF are not shown for L below 4 and 7, respectively, because for these values of L a high percentage of lost tracks (see Fig. 3(c)) renders

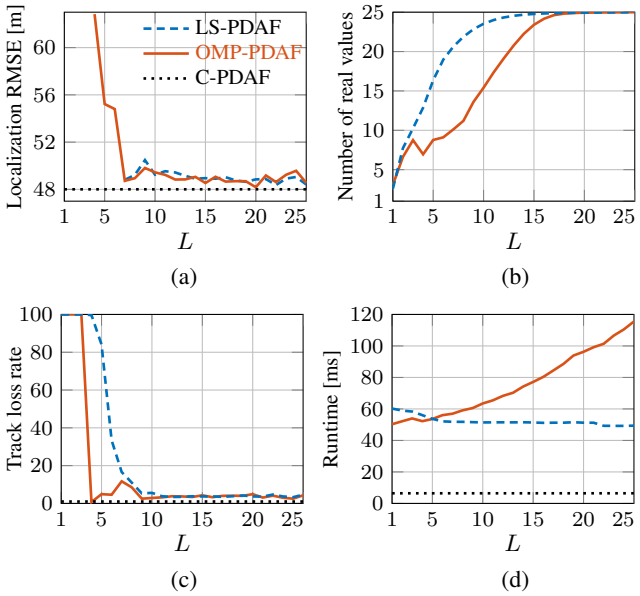

Fig. 3. Simulation results for different values of L: (a) Time-averaged localization RMSE, (b) time-averaged number of real values broadcast per sensor and consensus iteration, (c) TLR, (d) average runtime per time step.

the RMSE meaningless. It is seen in Fig. 3(a) that OMP-PDAF performs similar to, or slightly better than, LS-PDAF in the practically interesting L range between 7 and 12. For $L\geq 7$, OMP-PDAF and LS-PDAF perform quite close to C-PDAF. Fig. 3(b) depicts the average number of real values broadcast by one sensor during one consensus iteration, averaged over all times, consensus iterations, and sensors, versus L. For L between 4 and 14, OMP-PDAF is seen to require significantly less communication than LS-PDAF.

Fig. 3(c) shows the track loss rate (TLR), i.e., the percentage of simulation runs where a track loss occurred, versus L. The TLR of C-PDAF is always exactly zero. For $L\geq 7$, OMP-PDAF and LS-PDAF exhibit a similar TLR, which moreover is quite low for $L\geq 9$ (mostly about 4%). However, for L between 4 and 6, the TLR of OMP-PDAF is still reasonably low whereas that of LS-PDAF is high or very high. This means that OMP-PDAF allows for smaller values of L without incurring an excessive risk of track loss.

Finally, Fig. 3(d) displays the average runtime per time (filtering) step n versus L. In the practically interesting L range between 5 and 10, OMP-PDAF is slightly more complex than LS-PDAF, and for higher L it is significantly more complex.

We can conclude that for a certain range of the design parameter L (between 5 and 12), OMP-PDAF requires significantly less communication than LS-PDAF while exhibiting a similar or even better tracking performance and only a moderately higher complexity.

6. CONCLUSION

The proposed distributed particle-based probabilistic data association filter extends the likelihood consensus based distributed particle filter to scenarios with clutter and missed detections. A reduction of intersensor communication is achieved by using the orthogonal matching pursuit (OMP) for a sparse approximation of the local likelihood functions. Our simulation results show that the OMP leads to an attractive and flexible tradeoff between tracking performance and communication cost. The extension of the proposed method to multiple targets is an interesting direction for future research.

7. REFERENCES

[1] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, Yaakov Bar-Shalom, Storrs, CT, 2011.

[2] S. Challa, M. R. Morelande, D. Mušicki, and R. J. Evans, *Fundamentals of Object Tracking*, Cambridge University Press, New York, NY, 2011.

[3] F. Meyer, T. Kropfreiter, J. L. Williams, R. A. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proc. IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[4] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.

[5] G. Ferri, A. Munafò, A. Tesei, P. Braca, F. Meyer, K. Pelekanakis, R. Petroccia, J. Alves, C. Strode, and K. LePage, "Cooperative robotic networks for underwater surveillance: An overview," *IET Radar Sonar Navig.*, vol. 11, no. 12, pp. 1740–1761, 2017.

[6] 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, no. 5, pp. 19–38, Sept. 2003.

[7] 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, no. 2, pp. 174–188, Feb. 2002.

[8] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, San Francisco, CA, 2004.

[9] 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, no. 8, pp. 4334–4349, Aug. 2012.

[10] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Consensus-based distributed particle filtering with distributed proposal adaptation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, June 2014.

[11] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, Berlin, Germany, 2010.

[12] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

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

[14] T. Li, M. Bolic, and P. M. Djurić, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[16] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IPSN-05*, Boise, ID, Apr. 2005, pp. 63–70.

[17] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Feb. 2004.

[18] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[19] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[20] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, *Estimation with Applications to Tracking and Navigation*, Wiley, New York, NY, 2002.