A Distributed Bernoulli Filter Based on Likelihood Consensus with Adaptive Pruning

Rene Repp, Giuseppe Papa, Florian Meyer, Paolo Braca, and Franz Hlawatsch

***Abstract*—The Bernoulli filter (BF) is a Bayes-optimal method for target tracking when the target can be present or absent in unknown time intervals and the measurements are affected by clutter and missed detections. We propose a distributed particle-based multisensor BF algorithm that approximates the centralized multisensor BF for arbitrary nonlinear and non-Gaussian system models. Our distributed algorithm uses a new extension of the likelihood consensus (LC) scheme that accounts for both target presence and absence and includes an adaptive pruning of the LC expansion coefficients. Simulation results for a heterogeneous sensor network with significant noise and clutter show that the performance of our algorithm is close to that of the centralized multisensor BF.**

***Index Terms*—Bernoulli filter, distributed target tracking, distributed particle filtering, likelihood consensus, random finite set, sensor network.**

I. Introduction

The Bernoulli filter (BF) [1]–[3] is a Bayes-optimal method for detecting and tracking a single target that can be present or absent in arbitrary unknown time intervals. The BF is suited to measurements with clutter and missed detections. A multisensor extension of the BF and a particle-based implementation were proposed in [1]–[3].

In the multisensor case, a distributed, cooperative processing mode has numerous advantages over a centralized mode [4]: it does not require a fusion center, communication between distant points, or complex routing strategies; it is robust to network node and link failures; it is able to adapt to changing network topologies; and its complexity scales well with the network size. Two distributed BFs were proposed recently. In the *consensus BF* [5], single-sensor posterior probability density functions (pdfs) are transmitted to neighboring sensors and fused by means of a generalized covariance intersection technique [6]. The *random exchange BF*, which was introduced in [7] for received-signal-strength measurements, employs a diffusion technique in which each sensor swaps its local posterior pdf with that of a randomly chosen neighboring sensor and broadcasts its local measurements to all the neighboring sensors. These existing distributed BFs are based on the fusion of local posterior pdfs, which is not Bayes-optimal.

In this paper, we propose a distributed, particle-based, approximate implementation of the Bayes-optimal multisensor BF introduced in [2]. Our method is suited to arbitrary nonlinear and non-Gaussian measurement and state-evolution models. To the best of our knowledge, it is the first distributed multisensor BF that approximates the Bayes-optimal solution.

A distributed formulation of the centralized multisensor BF in [2] is complicated by the fact that in a decentralized sensor network, the joint likelihood function (JLF) is not available at the individual sensors. The proposed distributed BF uses the likelihood consensus (LC) scheme [8], [9] to provide an approximation of the JLF to each sensor, based on local communication between neighboring sensors. This approximate JLF is then used at the respective sensor in a local BF, whose performance approximates that of the centralized multisensor BF. To reduce the communication requirements of the LC, we propose an adaptive pruning through a thresholding in each consensus iteration.

This paper is organized as follows. The system model is described in Section II. Section III reviews the particle-based implementation of the centralized multisensor BF [1], [3]. Section IV develops an LC-based distributed calculation of the JLF with adaptive pruning. The proposed distributed BF is presented in Section V. Finally, Section VI reports simulation results demonstrating the performance and communication requirements of our method and comparing them with those of the random exchange BF [7].

II. System Model

We consider a network of S sensors, where each sensor $s \in \{1,\dots,S\}$ is able to communicate with a subset $\mathcal{S}_s \subseteq \{1,\dots,S\} \backslash \{s\}$ of the other sensors, termed the "neighbors" of sensor s. Communication is assumed symmetric, i.e., $s' \in \mathcal{S}_s$ implies $s \in \mathcal{S}_{s'}$. The communication graph—defined by all neighbor sets $\mathcal{S}_s$—is assumed to be connected.

A. Target State

The target may be absent or present at any given time $k \in \{1,2,\dots\}$. Accordingly, the target state at time k is modeled by a Bernoulli random finite set (RFS) X_k, which is empty (i.e., $\emptyset$) if the target is absent and $\{\boldsymbol{x}_k\}$ if the target is present [1], [10]. Here, in the latter case, $\boldsymbol{x}_k \in \mathbb{R}^d$ is a random state vector. The associated RFS pdf [1], [10] is

$$f(X_k) = \begin{cases} 1 - q_k, & X_k = \emptyset, \\ q_k f(\boldsymbol{x}_k), & X_k = \{\boldsymbol{x}_k\}, \end{cases} \tag{1}$$

R. Repp and F. Hlawatsch are with the Institute of Telecommunications, TU Wien, 1040 Vienna, Austria and with Brno University of Technology, 60190 Brno, Czech Republic (e-mail: rene.repp@nt.tuwien.ac.at, franz.hlawatsch@tuwien.ac.at). G. Papa is with Kognitiv Corporation, 1070 Vienna, Austria (e-mail: giuseppe.papa@kognitiv.com). F. Meyer is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: fmeyer@mit.edu). P. Braca is with the NATO STO Centre for Maritime Research and Experimentation (CMRE), 19126 La Spezia, Italy (e-mail: paolo.braca@cmre.nato.int). This work was supported by the NATO Supreme Allied Command Transformation under projects SAC000601 and SAC000608, and by the Austrian Science Fund (FWF) under projects P27370-N30 and J3886-N31.

where q_k is the target presence probability and $f(\boldsymbol{x}_k)$ is the pdf of the state vector. The state-transition pdf modeling the temporal evolution of the state X_k is given by [1]

$$f(X_k|X_{k-1}) = \begin{cases} 1-P_{\mathrm{b}}, & X_k=\emptyset,\, X_{k-1}=\emptyset, \\ P_{\mathrm{b}} f_{\mathrm{b}}(\boldsymbol{x}_k), & X_k=\{\boldsymbol{x}_k\},\, X_{k-1}=\emptyset, \\ 1-P_{\mathrm{s}}, & X_k=\emptyset,\, X_{k-1}=\{\boldsymbol{x}_{k-1}\}, \\ P_{\mathrm{s}} f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}), & X_k=\{\boldsymbol{x}_k\},\, X_{k-1}=\{\boldsymbol{x}_{k-1}\}, \end{cases}$$

where P_{b} is the birth probability, P_{s} is the survival probability, $f_{\mathrm{b}}(\boldsymbol{x}_k)$ is the birth pdf, and $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ describes the evolution of the state vector.

B. Measurements and Likelihood Functions

At time k, sensor $s \in \{1,\dots,S\}$ observes the set of measurements $Z_k^{(s)} \triangleq \{\boldsymbol{z}_{k,i}^{(s)}\}_{i=1}^{|Z_k^{(s)}|}$, where $|Z_k^{(s)}|$ denotes the cardinality of the set $Z_k^{(s)}$. Following [10], [11], we assume that either all the measurement vectors $\boldsymbol{z}_{k,i}^{(s)}$ are clutter-originated (aka false alarms), or one measurement vector is target-originated and all the others are clutter-originated. When the target is absent, only the first case is possible. When the target is present, the second case occurs with probability $P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)$ (known as the detection probability) and the first case with probability $1-P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)$. The clutter-originated measurements are modeled as independent and identically distributed (i.i.d.), as well as independent—both unconditionally and conditioned on the target states—of the target-originated measurement.

The measurement characteristic of sensor s that results from these assumptions is described by the *local likelihood function* (LLF) $f(Z_k^{(s)}|X_k)$, where X_k may be $\emptyset$ or $\{\boldsymbol{x}_k\}$. When $X_k=\emptyset$, all the measurements are clutter and $Z_k^{(s)}$ is an *i.i.d. cluster RFS* [1], whose pdf is given by

$$f(Z_k^{(s)}|X_k{=}\emptyset) = \tilde{f}_{\mathrm{c}}^{(s)}(Z_k^{(s)}) \triangleq |Z_k^{(s)}|!\, p_{\mathrm{c}}^{(s)}(|Z_k^{(s)}|) \prod_{\boldsymbol{z}\in Z_k^{(s)}} f_{\mathrm{c}}^{(s)}(\boldsymbol{z}). \quad (2)$$

Here, $p_{\mathrm{c}}^{(s)}(n)$ is the probability mass function (pmf) of the number of clutter measurements and $f_{\mathrm{c}}^{(s)}(\boldsymbol{z})$ is the pdf of each clutter measurement. For $Z_k^{(s)}=\emptyset$, Equation (2) reduces to $\tilde{f}_{\mathrm{c}}^{(s)}(\emptyset) = p_{\mathrm{c}}^{(s)}(0)$. When $X_k=\{\boldsymbol{x}_k\}$, $f(Z_k^{(s)}|X_k=\{\boldsymbol{x}_k\})$ is given by [1, Sec. V-A]

$$f(Z_k^{(s)}|X_k{=}\{\boldsymbol{x}_k\}) = \big(1-P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)\big)\tilde{f}_{\mathrm{c}}^{(s)}(Z_k^{(s)}) + P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k) \times \sum_{\boldsymbol{z}\in Z_k^{(s)}} f^{(s)}(\boldsymbol{z}|\boldsymbol{x}_k)\tilde{f}_{\mathrm{c}}^{(s)}(Z_k^{(s)}\backslash\{\boldsymbol{z}\}), \quad (3)$$

with some "vector likelihood function" $f^{(s)}(\boldsymbol{z}|\boldsymbol{x}_k)$. For $Z_k^{(s)}=\emptyset$, (3) reduces to $f(\emptyset|X_k=\{\boldsymbol{x}_k\}) = \big(1-P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)\big) \times\tilde{f}_{\mathrm{c}}^{(s)}(\emptyset) = \big(1-P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k)\big)p_{\mathrm{c}}^{(s)}(0)$.

Next, we combine the measurement sets $Z_k^{(s)}$ of all the sensors s into the ordered sequence $Z_k \triangleq (Z_k^{(s)})_{s=1}^S$, and we consider the *joint likelihood function* (JLF) $f(Z_k|X_k)$. Since the measurements at the individual sensors are assumed conditionally independent given the target state X_k, the JLF is given by

$$f(Z_k|X_k) = \prod_{s=1}^S f(Z_k^{(s)}|X_k). \quad (4)$$

III. Review of Particle-Based Bernoulli Filtering

The proposed distributed BF is based algorithmically on the particle-based implementation of the centralized multisensor BF [1], [3]. For a review of that method, we assume that the all-sensors measurement Z_k is available at a central processing unit. We denote by $Z_{1:k} \triangleq (Z_i)_{i=1}^k$ the sequence of the all-sensors measurements up to the current time k.

The centralized multisensor BF is a Bayesian detection/estimation method that calculates at each time k the posterior pdf of the target state X_k given $Z_{1:k}$, $f(X_k|Z_{1:k})$ [1]. This posterior pdf is of the Bernoulli form (1), i.e.,

$$f(X_k|Z_{1:k}) \triangleq \begin{cases} 1-q_{k|k}, & X_k=\emptyset, \\ q_{k|k} f(\boldsymbol{x}_k|Z_{1:k}), & X_k=\{\boldsymbol{x}_k\}, \end{cases}$$

where $q_{k|k} \triangleq \Pr(X_k\neq\emptyset|Z_{1:k})$ is the posterior probability of target presence and $f(\boldsymbol{x}_k|Z_{1:k})$ is the posterior pdf of the state vector $\boldsymbol{x}_k$. In the particle-based implementation of the centralized multisensor BF [1], [3], $f(X_k|Z_{1:k})$ is represented by an approximate target presence probability $\hat{q}_{k|k}$ and a set $\big\{\big(\boldsymbol{x}_k^{(i)}, w_k^{(i)}\big)\big\}_{i=1}^{I_{\mathrm{p}}}$ of I_{p} particles $\boldsymbol{x}_k^{(i)}$ and associated weights $w_k^{(i)}$ that provide an approximate representation of $f(\boldsymbol{x}_k|Z_{1:k})$. Both $\hat{q}_{k|k}$ and $\big\{\big(\boldsymbol{x}_k^{(i)}, w_k^{(i)}\big)\big\}_{i=1}^{I_{\mathrm{p}}}$ are calculated in a time-recursive manner via a prediction step and an update step.

A. Prediction Step

In the prediction step [1], the previous posterior target presence probability $\hat{q}_{k-1|k-1}$ is converted into a predicted target presence probability $\hat{q}_{k|k-1}$ according to

$$\hat{q}_{k|k-1} = P_{\mathrm{b}}(1-\hat{q}_{k-1|k-1}) + P_{\mathrm{s}}\hat{q}_{k-1|k-1}. \quad (5)$$

Furthermore, the previous particle representation $\big\{\big(\boldsymbol{x}_{k-1}^{(i)}, w_{k-1}^{(i)}\big)\big\}_{i=1}^{I_{\mathrm{p}}}$ is converted into a predicted particle representation $\big\{\big(\boldsymbol{x}_{k|k-1}^{(i)}, w_{k|k-1}^{(i)}\big)\big\}_{i=1}^{I_{\mathrm{p}}+I_{\mathrm{b}}}$, where I_{b} is the number of "birth particles." Here, the particles $\boldsymbol{x}_{k|k-1}^{(i)}$ are sampled for $i=1,\dots,I_{\mathrm{p}}$ from the vector state-transition pdf $f\big(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}{=}\boldsymbol{x}_{k-1}^{(i)}\big)$ and for $i=I_{\mathrm{p}}{+}1,\dots,I_{\mathrm{p}}{+}I_{\mathrm{b}}$ from the birth pdf $f_{\mathrm{b}}(\boldsymbol{x}_k)$, and the weights are obtained as

$$w_{k|k-1}^{(i)} = \begin{cases} \dfrac{P_{\mathrm{s}}\hat{q}_{k-1|k-1}}{\hat{q}_{k|k-1}} w_{k-1}^{(i)}, & i=1,\dots,I_{\mathrm{p}} \\ \dfrac{1}{I_{\mathrm{b}}} \dfrac{P_{\mathrm{b}}(1-\hat{q}_{k-1|k-1})}{\hat{q}_{k|k-1}}, & i=I_{\mathrm{p}}{+}1,\dots,I_{\mathrm{p}}{+}I_{\mathrm{b}}. \end{cases} \quad (6)$$

B. Update Step

In the update step [1], $\hat{q}_{k|k-1}$ is converted into the new posterior target presence probability $\hat{q}_{k|k}$ according to

$$\hat{q}_{k|k} = 1 - \frac{f(Z_k|X_k{=}\emptyset)(1-\hat{q}_{k|k-1})}{\hat{f}(Z_k|Z_{1:k-1})}. \quad (7)$$

Here,

$$\hat{f}(Z_k|Z_{1:k-1}) = C(Z_{1:k})\,\hat{q}_{k|k-1} + f(Z_k|X_k{=}\emptyset)\,(1-\hat{q}_{k|k-1})\,, \tag{8}$$

with

$$C(Z_{1:k}) \triangleq \sum_{i=1}^{I_\text{p}+I_\text{b}} f\big(Z_k\big|X_k{=}\{\boldsymbol{x}_{k|k-1}^{(i)}\}\big)\,w_{k|k-1}^{(i)}\,. \tag{9}$$

Then, $\big\{\big(\boldsymbol{x}_{k|k-1}^{(i)}, w_{k|k-1}^{(i)}\big)\big\}_{i=1}^{I_\text{p}+I_\text{b}}$ is converted into the new particle representation $\big\{\big(\boldsymbol{x}_k^{(i)}, w_k^{(i)}\big)\big\}_{i=1}^{I_\text{p}}$. First, intermediate weights are calculated as

$$\tilde{w}_k^{(i)} = f\big(Z_k\big|X_k{=}\{\boldsymbol{x}_{k|k-1}^{(i)}\}\big)\,w_{k|k-1}^{(i)}\,, \quad i=1,\dots,I_\text{p}+I_\text{b}\,. \tag{10}$$

Next, from the predicted particle set $\big\{\boldsymbol{x}_{k|k-1}^{(i)}\big\}_{i=1}^{I_\text{p}+I_\text{b}}$, only the I_p particles $\boldsymbol{x}_{k|k-1}^{(i)}$ with the highest weights $\tilde{w}_k^{(i)}$ are retained, and these weights are normalized such that their sum is 1. Denoting the retained particles as $\boldsymbol{x}_k^{(i)}$ and the corresponding normalized weights as $w_k^{(i)}$, with a reindexing such that $i \in \{1,\dots,I_\text{p}\}$, we have now obtained a set of particles and normalized weights $\big\{\big(\boldsymbol{x}_k^{(i)}, w_k^{(i)}\big)\big\}_{i=1}^{I_\text{p}}$; this set provides an approximate representation of $f(\boldsymbol{x}_k|Z_{1:k})$. Finally, to avoid particle degeneracy, a resampling step is executed if the *effective sample size* [12] is below a given threshold.

C. Initialization. Detection and Estimation

The recursion constituted by the above prediction and update steps is initialized at time $k=1$ by setting $\hat{q}_{0|0}$ equal to some initial prior target presence probability q_0 (e.g., 0.5), drawing the particles $\big\{\boldsymbol{x}_0^{(i)}\big\}_{i=1}^{I_\text{p}}$ from an initial prior state vector pdf $f(\boldsymbol{x}_0)$ (e.g., uniform on the surveillance region), and using equal initial weights $\big\{w_0^{(i)} \equiv 1/I_\text{p}\big\}_{i=1}^{I_\text{p}}$.

At each time step k, the target is detected (i.e., declared to be present) if $\hat{q}_{k|k}$ is above a threshold P_th [13, Ch. 2]. In that case, an estimate of the target state $\boldsymbol{x}_k$ is calculated from the particle set $\big\{\big(\boldsymbol{x}_k^{(i)}, w_k^{(i)}\big)\big\}_{i=1}^{I_\text{p}}$ representing $f(\boldsymbol{x}_k|Z_{1:k})$ according to $\hat{\boldsymbol{x}}_k = \sum_{i=1}^{I_\text{p}} w_k^{(i)}\boldsymbol{x}_k^{(i)}$. This estimate approximates the minimum mean-square error estimator $\hat{\boldsymbol{x}}_k^\text{MMSE} \triangleq \int \boldsymbol{x}_k f(\boldsymbol{x}_k|Z_{1:k})\,\mathrm{d}\boldsymbol{x}_k$ [13, Ch. 4].

IV. Distributed Calculation of the JLF

In the update step, the JLF $f(Z_k|X_k)$ is needed: $f(Z_k|X_k{=}\emptyset)$ is involved in (7) and (8), and $f\big(Z_k\big|X_k = \{\boldsymbol{x}_{k|k-1}^{(i)}\}\big)$ for $i=1,\dots,I_\text{p}+I_\text{b}$ in (9) and (10). In a distributed implementation, each sensor s runs its own local instance of a BF, briefly termed *local BF* in the following. Hence, if the local BFs are desired to perform similarly to the centralized BF, then sufficiently accurate approximations of $f(Z_k|X_k{=}\emptyset)$ and $f\big(Z_k\big|X_k{=}\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i=1,\dots,I_\text{p}{+}I_\text{b}$ must be available at each sensor. Here, $\boldsymbol{x}_{k|k-1}^{(s,i)}$ denotes the ith predicted particle used by the local BF at sensor s; note that the particles at different sensors are generally different.

However, each sensor s knows only its own local measurement $Z_k^{(s)}$ and, thus, its own LLF $f(Z_k^{(s)}|X_k)$. Therefore, the core of the proposed distributed BF is a distributed scheme for calculating approximations of $f(Z_k|X_k{=}\emptyset)$ and $f\big(Z_k\big|X_k{=}\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i=1,\dots,I_\text{p}+I_\text{b}$ using a consensus algorithm in the first case and the LC in the second case. This calculation uses the predicted particle representation $\big\{\big(\boldsymbol{x}_{k|k-1}^{(s,i)}, w_{k|k-1}^{(s,i)}\big)\big\}_{i=1}^{m}$, with $m \triangleq I_\text{p}+I_\text{b}$, and it employs only local communication between neighboring sensors.

The starting-point for developing the distributed algorithm is to write the JLF as

$$f(Z_k|X_k) = \exp\big(S L_k(X_k)\big)\,, \tag{11}$$

with

$$L_k(X_k) \triangleq \frac{1}{S}\log f(Z_k|X_k) = \frac{1}{S}\sum_{s=1}^{S}\log f(Z_k^{(s)}|X_k)\,, \tag{12}$$

where log denotes the natural logarithm and (4) was used in the last step. (Note that our notation does not show the dependence of $L_k(X_k)$ on Z_k.) Thus, we are now seeking a distributed calculation of $L_k(X_k)$ under both target absence, $X_k=\emptyset$, and target presence, $X_k=\{\boldsymbol{x}_k\}$.

A. Calculation of $f(Z_k|X_k{=}\emptyset)$

For $X_k{=}\emptyset$, (12) becomes

$$L_k(\emptyset) = \frac{1}{S}\sum_{s=1}^{S}\log f(Z_k^{(s)}|X_k{=}\emptyset)\,. \tag{13}$$

A distributed calculation of this average can be performed iteratively by the *average consensus algorithm* [14]. In iteration $j \in \{1,2,\dots\}$, sensor s calculates the "internal state"

$$u_j^{(s)} = \sum_{s' \in \{s\}\cup\mathcal{S}_s} \gamma_{s,s'}\,u_{j-1}^{(s')}\,, \tag{14}$$

where the $\gamma_{s,s'}$ are suitably chosen weights (e.g., the Metropolis weights [15]) and the $u_{j-1}^{(s')}$ were calculated by sensor s and its neighbors $s' \in \mathcal{S}_s$ in the previous iteration $j-1$. The recursion (14) is initialized by $u_0^{(s)} = \log f(Z_k^{(s)}|X_k{=}\emptyset)$. Note that this scheme requires that in each consensus iteration j, each sensor s broadcasts its previous internal state $u_{j-1}^{(s)}$ to its neighbors $s' \in \mathcal{S}_s$. For $j\to\infty$, $u_j^{(s)}$ is guaranteed to converge to $L_k(\emptyset)$ in (13) [15]. For a finite number J of iterations, $u_J^{(s)}$ provides only an approximation of $L_k(\emptyset)$, i.e., $L_k(\emptyset) \approx u_J^{(s)}$. Inserting this approximation into (11) then yields a corresponding approximation of the JLF $f(Z_k|X_k{=}\emptyset)$,

$$f(Z_k|X_k{=}\emptyset) \approx \hat{f}^{(s)}(Z_k|X_k{=}\emptyset) \triangleq \exp\big(S u_J^{(s)}\big)\,.$$

In practice, the number of iterations J is either specified beforehand or chosen adaptively by terminating the contribution of sensor s to the consensus iterations when $\big|u_j^{(s)} - u_{j-1}^{(s)}\big|$ falls below a threshold.

B. Calculation of $f\big(Z_k\big|X_k{=}\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$

For $X_k{=}\{\boldsymbol{x}_k\}$, (12) becomes

$$L_k(\{\boldsymbol{x}_k\}) = \frac{1}{S}\sum_{s=1}^{S}\log f(Z_k^{(s)}|X_k{=}\{\boldsymbol{x}_k\})\,. \tag{15}$$

This is the arithmetic average of *functions* of $\boldsymbol{x}_k \in \mathbb{R}^d$, rather than of numbers as in (13), and thus it cannot be directly calculated by the average consensus algorithm. The idea of the

LC scheme, which was first proposed for distributed particle filtering [8], [9], is to expand the log-LLFs $\log f(Z_k^{(s)}|X_k=\{\boldsymbol{x}_k\})$ into a dictionary of functions and then use the average consensus algorithm on each expansion coefficient.

1) Review of the Likelihood Consensus: We consider the approximation of the log-LLF at sensor s by a finite-order function expansion, i.e. [9]

$$\log f(Z_k^{(s)}|X_k=\{\boldsymbol{x}_k\}) \approx \sum_{b=1}^{B} \alpha_{k,b}^{(s)}\psi_b(\boldsymbol{x}_k), \tag{16}$$

where $\{\psi_b(\boldsymbol{x}_k)\}_{b=1}^{B}$ is a fixed dictionary of functions that is identical at all the sensors and known to each sensor. Here, in particular, we will use a d-D Fourier dictionary (recall that $\boldsymbol{x}_k \in \mathbb{R}^d$), which is well suited to the multicomponent, continuous local LLFs arising in Bernoulli filtering. More specifically, let us consider a d-D surveillance region $\mathcal{R} = [0,D_1] \times \cdots \times [0,D_d]$, i.e., we assume that $\boldsymbol{x}_k \in \mathcal{R}$. The atoms of the d-D Fourier dictionary are then given by [9] $\psi_b(\boldsymbol{x}_k) = \prod_{\ell=1}^{d} \phi_{b'_\ell}^{(\ell)}(x_{k,\ell})$, $b = 1,\dots,B$, where $x_{k,\ell}$ denotes the ℓth entry of $\boldsymbol{x}_k$ and the 1-D atoms $\phi_{b'}^{(\ell)}(x)$ are given by

$$\phi_{b'}^{(\ell)}(x) = \begin{cases} 1, & b'=1 \\ \cos\big(\frac{2\pi}{D_\ell}(b'-1)x\big), & b'=2,\dots,B'_\ell+1 \\ \sin\big(\frac{2\pi}{D_\ell}(b'-1-B'_\ell)x\big), & b'=B'_\ell+2,\dots,2B'_\ell+1, \end{cases}$$

with a suitable index transformation relating the 1-D index $b \in \{1,\dots,B\}$ and the d-D index $(b'_1,\dots,b'_d) \in \{1,\dots,2B'_1+1\} \times \cdots \times \{1,\dots,2B'_d+1\}$. Note that the size of the d-D dictionary is $B = \prod_{\ell=1}^{d}(2B'_\ell+1)$.

The expansion coefficients $\{\alpha_{k,b}^{(s)}\}_{b=1}^{B}$ in (16) depend on the local measurement $Z_k^{(s)}$ and are thus generally different at different sensors s. At each sensor s, the local coefficient vector $\boldsymbol{\alpha}_k^{(s)} \triangleq (\alpha_{k,1}^{(s)} \cdots \alpha_{k,B}^{(s)})^{\mathrm{T}}$ involved in (16) can be calculated *locally* via a least-squares (LS) fit [16], [17] of $\sum_{b=1}^{B}\alpha_{k,b}^{(s)}\psi_b(\boldsymbol{x}_k)$ to the log-LLF $\log f(Z_k^{(s)}|X_k = \{\boldsymbol{x}_k\})$. Because only $f\big(Z_k|X_k=\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i=1,\dots,m$ is needed in the update step of the local BF, the approximation of the log-LLF has to be good only in the state-space regions containing the predicted particles $\boldsymbol{x}_{k|k-1}^{(s,i)}$. Therefore, the LS fit minimizes, with respect to $\boldsymbol{\alpha}_k^{(s)}$, the approximation error $\sum_{i=1}^{m}\epsilon_i^2$ with $\epsilon_i \triangleq \log f\big(Z_k^{(s)}|X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big) - \sum_{b=1}^{B}\alpha_{k,b}^{(s)}\psi_b\big(\boldsymbol{x}_{k|k-1}^{(s,i)}\big)$. Assuming $m \geq B$ (i.e., there are at least as many particles as expansion coefficients), the solution to this minimization problem is given by [16]

$$\boldsymbol{\alpha}_k^{(s)} = \boldsymbol{\Psi}_k^{(s)\#}\boldsymbol{\lambda}_k^{(s)}, \quad \text{where } \boldsymbol{\Psi}_k^{(s)\#} \triangleq \big(\boldsymbol{\Psi}_k^{(s)\mathrm{T}}\boldsymbol{\Psi}_k^{(s)}\big)^{-1}\boldsymbol{\Psi}_k^{(s)\mathrm{T}}, \tag{17}$$

with the log-LLF vector $\boldsymbol{\lambda}_k^{(s)} \triangleq \Big(\log f\big(Z_k^{(s)}\big|X_k=\{\boldsymbol{x}_{k|k-1}^{(s,1)}\}\big) \cdots \log f\big(Z_k^{(s)}\big|X_k = \{\boldsymbol{x}_{k|k-1}^{(s,m)}\}\big)\Big)^{\mathrm{T}}$ and the $m \times B$ dictionary matrix $\boldsymbol{\Psi}_k^{(s)}$ whose columns are $\boldsymbol{\psi}_b^{(s)} \triangleq \big(\psi_b\big(\boldsymbol{x}_{k|k-1}^{(s,1)}\big) \cdots \psi_b\big(\boldsymbol{x}_{k|k-1}^{(s,m)}\big)\big)^{\mathrm{T}}$, $b = 1,\dots,B$. We note that the existence of $\big(\boldsymbol{\Psi}_k^{(s)\mathrm{T}}\boldsymbol{\Psi}_k^{(s)}\big)^{-1}$ in (17) presupposes that $\boldsymbol{\Psi}_k^{(s)}$ has full rank, i.e., the columns $\boldsymbol{\psi}_b^{(s)}$ are linearly independent. Numerical aspects of computing $\boldsymbol{\alpha}_k^{(s)}$ in (17) are discussed in [16], [17].

Inserting (16) into (15) and changing the order of summations yields the following approximation of the log-JLF $L_k(\{\boldsymbol{x}_k\})$:

$$L_k(\{\boldsymbol{x}_k\}) \approx \sum_{b=1}^{B} \beta_{k,b}\psi_b(\boldsymbol{x}_k), \tag{18}$$

with the global expansion coefficients

$$\beta_{k,b} \triangleq \frac{1}{S}\sum_{s=1}^{S}\alpha_{k,b}^{(s)}, \quad b = 1,\dots,B.$$

Thus, the calculation of $L_k(\{\boldsymbol{x}_k\})$ amounts to calculating the averages $\beta_{k,b}$ for $b = 1,\dots,B$. This can be done by executing B parallel instances of the average consensus algorithm. That is, sensor s updates B internal states $v_j^{(s,b)}$, $b = 1,\dots,B$ in parallel using the recursion (cf. (14))

$$v_j^{(s,b)} = \sum_{s' \in \{s\}\cup\mathcal{S}_s} \gamma_{s,s'}\, v_{j-1}^{(s',b)}. \tag{19}$$

This recursion is initialized by $v_0^{(s,b)} = \alpha_{k,b}^{(s)}$. After a sufficient number J of consensus iterations, the final internal states $v_J^{(s,b)}$ provide approximations of the $\beta_{k,b}$, i.e., $\beta_{k,b} \approx v_J^{(s,b)}$. Hence, (18) yields

$$L_k(\{\boldsymbol{x}_k\}) \approx \sum_{b=1}^{B} v_J^{(s,b)}\psi_b(\boldsymbol{x}_k). \tag{20}$$

Finally, inserting (20) into (11) and evaluating the resulting expression at $\boldsymbol{x}_k = \boldsymbol{x}_{k|k-1}^{(s,i)}$ yields the following approximation:

$$\begin{aligned} f\big(Z_k|X_k=\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big) &\approx \hat{f}^{(s)}\big(Z_k|X_k=\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big) \\ &\triangleq \exp\Bigg(S\sum_{b=1}^{B} v_J^{(s,b)}\psi_b\big(\boldsymbol{x}_{k|k-1}^{(s,i)}\big)\Bigg), \end{aligned}$$

for $i=1,\dots,m$.

2) Adaptive Pruning: In the distributed calculation of $f(Z_k|X_k = \emptyset)$ and $f\big(Z_k|X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i = 1,\dots,m$ according to Sections IV-A and IV-B1, sensor s executes $1+B$ parallel instances of the average consensus algorithm. In each consensus iteration $j \in \{1,\dots,J\}$, it broadcasts the $1+B$ internal states $u_{j-1}^{(s)}$ and $v_{j-1}^{(s,b)}$, $b = 1,\dots,B$ to its neighbors $s' \in \mathcal{S}_s$. To reduce this communication cost, we now propose an adaptive pruning of the internal states.

According to (16), at each sensor s, the log-LLF $\log f(Z_k^{(s)}|X_k=\{\boldsymbol{x}_k\})$ is represented by the local expansion coefficients $\alpha_{k,b}^{(s)}$, $b = 1,\dots,B$. We call $\alpha_{k,b}^{(s)}$ *active* if $|\alpha_{k,b}^{(s)}| > \eta_0$ with a suitably chosen positive threshold η_0, and we define the *active index set* $\mathcal{B}_0^{(s)} \triangleq \{b\colon |\alpha_{k,b}^{(s)}| > \eta_0\}$. Let us now modify the LC as follows. In the first consensus iteration ($j=1$), sensor s broadcasts to its neighbors $s' \in \mathcal{S}_s$ only its active coefficients, i.e., the initial internal states $v_0^{(s,b)} = \alpha_{k,b}^{(s)}$ only for $b \in \mathcal{B}_0^{(s)}$, and it receives from its neighbors $s' \in \mathcal{S}_s$ their initial internal states $v_0^{(s',b)}$ only for $b \in \mathcal{B}_0^{(s')}$. Sensor s considers the remaining (nonactive) initial internal states, $v_0^{(s',b)}$ for $b \notin \mathcal{B}_0^{(s')}$ and $s' \in \{s\}\cup\mathcal{S}_s$, as zero. The consensus update in (19) then yields the new internal states at sensor s as

$$v_1^{(s,b)} = \sum_{s' \in \{s\} \cup \mathcal{S}_s} \gamma_{s,s'} \, v_0^{(s',b)}, \quad b \in \tilde{\mathcal{B}}_1^{(s)},$$

where $\tilde{\mathcal{B}}_1^{(s)}$ is the union of all the involved active index sets, i.e., $\tilde{\mathcal{B}}_1^{(s)} \triangleq \bigcup_{s' \in \{s\} \cup \mathcal{S}_s} \mathcal{B}_0^{(s')}$. Sensor s now retains only those $v_1^{(s,b)}$ that satisfy $|v_1^{(s,b)}| > \eta_1$ with a suitably chosen positive threshold η_1; these are its new "active" internal states. This corresponds to a new active index set $\mathcal{B}_1^{(s)} \triangleq \big\{ b \in \tilde{\mathcal{B}}_1^{(s)} : |v_1^{(s,b)}| > \eta_1 \big\}$.

This pruning scheme is also used in the remaining consensus iterations. More precisely, in consensus iteration $j \in \{2, 3, \ldots, J\}$, sensor s uses a thresholding to determine its active internal states $v_{j-1}^{(s,b)}$, $b \in \mathcal{B}_{j-1}^{(s)}$, with the active index set $\mathcal{B}_{j-1}^{(s)} \triangleq \big\{ b \in \tilde{\mathcal{B}}_{j-1}^{(s)} : |v_{j-1}^{(s,b)}| > \eta_{j-1} \big\}$. Then, sensor s broadcasts to its neighbors $s' \in \mathcal{S}_s$ its active internal states $v_{j-1}^{(s,b)}$, $b \in \mathcal{B}_{j-1}^{(s)}$ and receives from them their active internal states $v_{j-1}^{(s',b)}$, $b \in \mathcal{B}_{j-1}^{(s')}$; it again considers all nonactive internal states, $v_{j-1}^{(s',b)}$ for $b \notin \mathcal{B}_{j-1}^{(s')}$ and $s' \in \{s\} \cup \mathcal{S}_s$, as zero. Finally, it calculates

$$v_j^{(s,b)} = \sum_{s' \in \{s\} \cup \mathcal{S}_s} \gamma_{s,s'} \, v_{j-1}^{(s',b)}, \quad b \in \tilde{\mathcal{B}}_j^{(s)},$$

with $\tilde{\mathcal{B}}_j^{(s)} \triangleq \bigcup_{s' \in \{s\} \cup \mathcal{S}_s} \mathcal{B}_{j-1}^{(s')}$.

A simple adaptive choice of the pruning thresholds η_j that has been observed to perform well is to set $\eta_j = \omega \cdot \frac{1}{B} \sum_{b=1}^{B} |\alpha_{k,b}^{(s)}|$ with a suitably chosen fixed factor $\omega > 0$. In addition, we observed that it is advantageous to start pruning only after a certain number of initial consensus iterations; this formally corresponds to initially using $\eta_j = 0$.

The communication cost of this *pruned LC* scheme is as follows. In consensus iteration j of the $1{+}B$ parallel consensus algorithms, sensor s broadcasts to its neighbors the internal states $u_{j-1}^{(s)}$ and $v_{j-1}^{(s,b)}$ for $b \in \mathcal{B}_{j-1}^{(s)}$, as well as the $|\mathcal{B}_{j-1}^{(s)}|$ indices $b \in \mathcal{B}_{j-1}^{(s)} \subseteq \{1, \ldots, B\}$. This amounts to $1 + |\mathcal{B}_{j-1}^{(s)}|$ real values plus $|\mathcal{B}_{j-1}^{(s)}|$ integer values taken from $\{1, \ldots, B\}$. Considering these integer values as real values for simplicity (although in fact they can be represented by a moderate number of bits), the total number of real values broadcast at each time k by sensor s to its neighbors during all the J consensus iterations is obtained as

$$N_{\mathrm{c}} = \sum_{j=1}^{J} \big(1 + 2|\mathcal{B}_{j-1}^{(s)}|\big) = J + 2 \sum_{j=1}^{J} |\mathcal{B}_{j-1}^{(s)}|.$$

C. Algorithm Summary

The overall consensus and pruned LC scheme for distributed calculation of $f(Z_k | X_k = \emptyset)$ and $f\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i = 1, \ldots, m$ is summarized in Algorithm 1.

V. LC-Based Distributed Bernoulli Filtering

We now combine the particle-based implementation of the centralized BF reviewed in Section III with the consensus-based distributed calculation of the JLF presented in Section IV. The resulting distributed implementation of the BF is stated in Algorithm 2. Each sensor s runs a local BF that is based on the JLF $f(Z_k | X_k)$. The local BF at sensor s calculates in a time-recursive manner a local estimate $\hat{q}_{k|k}^{(s)}$ of the target presence probability $q_{k|k}$ and local particle representations $\big\{ \big(\boldsymbol{x}_k^{(s,i)}, w_k^{(s,i)} \big) \big\}_{i=1}^{I_{\mathrm{p}}}$ of the posterior state vector pdf $f(\boldsymbol{x}_k | Z_{1:k})$ (cf. Section III-B). The calculations of $\hat{q}_{k|k}^{(s)}$ and $\big\{ \big(\boldsymbol{x}_k^{(s,i)}, w_k^{(s,i)} \big) \big\}_{i=1}^{I_{\mathrm{p}}}$ at different sensors s are done separately but they involve approximations of the JLF and not merely the LLFs of the respective sensors s. Thus, each local BF uses the measurements of the entire sensor network and, thereby, is able to perform similarly to the centralized BF.

Algorithm 1 Consensus scheme for distributed JLF calculation—Operations performed by sensor s at time $k \in \{1, 2, \ldots\}$

```
Input: $\{\boldsymbol{x}_{k|k-1}^{(s,i)}\}_{i=1}^{m}$ and $Z_k^{(s)}$;
Calculate $u_0^{(s)} = \log f(Z_k^{(s)} | X_k = \emptyset)$ using (2);
Form $\boldsymbol{\Psi}_k^{(s)}$ and $\boldsymbol{\lambda}_k^{(s)}$ as discussed in Section IV-B1;
Calculate $\boldsymbol{\alpha}_k^{(s)}$ using (17);
Set $v_0^{(s,b)} = \alpha_{k,b}^{(s)}$ for $b = 1, \ldots, B$ and $\tilde{\mathcal{B}}_0^{(s)} = \{1, \ldots, B\}$;
for $j = 1$ to $J$ do
    Determine the active index set $\mathcal{B}_{j-1}^{(s)} = \{ b \in \tilde{\mathcal{B}}_{j-1}^{(s)} : |v_{j-1}^{(s,b)}| > \eta_{j-1} \}$;
    Broadcast $u_{j-1}^{(s)}$ as well as the active internal states $v_{j-1}^{(s,b)}$, $b \in \mathcal{B}_{j-1}^{(s)}$ and their indices $b \in \mathcal{B}_{j-1}^{(s)}$ to the neighbor sensors $s' \in \mathcal{S}_s$; receive $u_{j-1}^{(s')}$ as well as the active internal states $v_{j-1}^{(s',b)}$, $b \in \mathcal{B}_{j-1}^{(s')}$ and their indices $b \in \mathcal{B}_{j-1}^{(s')}$ from the neighbor sensors $s' \in \mathcal{S}_s$;
    Calculate $u_j^{(s)} = \sum_{s' \in \{s\} \cup \mathcal{S}_s} \gamma_{s,s'} u_{j-1}^{(s')}$;
    Form $\tilde{\mathcal{B}}_j^{(s)} = \bigcup_{s' \in \{s\} \cup \mathcal{S}_s} \mathcal{B}_{j-1}^{(s')}$;
    for $b \in \tilde{\mathcal{B}}_j^{(s)}$ do
        Calculate $v_j^{(s,b)} = \sum_{s' \in \{s\} \cup \mathcal{S}_s} \gamma_{s,s'} v_{j-1}^{(s',b)}$, where nonactive $v_{j-1}^{(s',b)}$ are considered to be zero;
    end for
end for
Calculate an approximation of $f(Z_k | X_k = \emptyset)$ as $\hat{f}^{(s)}(Z_k | X_k = \emptyset) = \exp\big(S u_J^{(s)}\big)$;
for $i = 1$ to $m$ do
    Calculate an approximation of $f\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$ as $\hat{f}^{(s)}\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big) = \exp\Big(S \sum_{b=1}^{B} v_J^{(s,b)} \psi_b\big(\boldsymbol{x}_{k|k-1}^{(s,i)}\big)\Big)$;
end for
Output: $\hat{f}^{(s)}(Z_k | X_k = \emptyset)$ and $\hat{f}^{(s)}\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i = 1, \ldots, m$.
```

The main difference of Algorithm 2 from the centralized BF reviewed in Section III is Step 10. In this step, the LC—i.e., Algorithm 1—is used to provide sensor s with $\hat{f}^{(s)}(Z_k | X_k = \emptyset)$ and $\hat{f}^{(s)}\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$, $i = 1, \ldots, I_{\mathrm{p}} + I_{\mathrm{b}}$, which are then used instead of $f(Z_k | X_k = \emptyset)$ and $f\big(Z_k \big| X_k = \{\boldsymbol{x}_{k|k-1}^{(s,i)}\}\big)$ in the subsequent steps of the filter. Note that because each local BF uses its own set of local particles, the random generation of particles from the local proposal pdfs $f\big(\boldsymbol{x}_k \big| \boldsymbol{x}_{k-1} = \boldsymbol{x}_{k-1}^{(s,i)}\big)$ and $f_{\mathrm{b}}(\boldsymbol{x}_k)$ does not have to be synchronized across the different sensors s.

Algorithm 2 Distributed Bernoulli filter—operations performed by sensor s at time $k \in \{1,2,\dots\}$

```
Input: $\hat{q}^{(s)}_{k-1|k-1}$, $\big\{\big(\boldsymbol{x}^{(s,i)}_{k-1}, w^{(s,i)}_{k-1}\big)\big\}_{i=1}^{I_{\mathrm{p}}}$, and $Z^{(s)}_k$
    (Initialization: $\hat{q}^{(s)}_{0|0}$ is set to the initial prior target presence probability $q_0$, the particles $\{\boldsymbol{x}^{(s,i)}_0\}_{i=1}^{I_{\mathrm{p}}}$ are drawn from the initial prior state vector pdf $f(\boldsymbol{x}_0)$, and the weights are chosen equal, i.e., $w^{(s,i)}_0 \equiv 1/I_{\mathrm{p}}$)
Calculate $\hat{q}^{(s)}_{k|k-1}$ using (5), i.e.,
    $\hat{q}^{(s)}_{k|k-1} = P_{\mathrm{b}}\big(1-\hat{q}^{(s)}_{k-1|k-1}\big) + P_{\mathrm{s}}\,\hat{q}^{(s)}_{k-1|k-1}\,;$
for $i=1$ to $I_{\mathrm{p}}$ do
    For each particle $\boldsymbol{x}^{(s,i)}_{k-1}$, draw a predicted particle $\boldsymbol{x}^{(s,i)}_{k|k-1}$ from the state-transition pdf $f\big(\boldsymbol{x}_k\big|\boldsymbol{x}_{k-1}=\boldsymbol{x}^{(s,i)}_{k-1}\big)$;
    Calculate a corresponding weight according to (6), i.e.,
        $w^{(s,i)}_{k|k-1} = \dfrac{P_{\mathrm{s}}\,\hat{q}^{(s)}_{k-1|k-1}}{\hat{q}^{(s)}_{k|k-1}}\, w^{(s,i)}_{k-1}\,;$
end for
for $i=I_{\mathrm{p}}+1$ to $I_{\mathrm{p}}+I_{\mathrm{b}}$ do
    Draw a birth particle $\boldsymbol{x}^{(s,i)}_{k|k-1}$ from the birth pdf $f_{\mathrm{b}}(\boldsymbol{x}_k)$;
    Calculate a corresponding weight according to (6), i.e.,
        $w^{(s,i)}_{k|k-1} = \dfrac{1}{I_{\mathrm{b}}}\,\dfrac{P_{\mathrm{b}}\big(1-\hat{q}^{(s)}_{k-1|k-1}\big)}{\hat{q}^{(s)}_{k|k-1}}\,;$
end for
Execute Algorithm 1 with $m = I_{\mathrm{p}}+I_{\mathrm{b}}$ to obtain $\hat{f}^{(s)}(Z_k|X_k=\emptyset)$ and $\hat{f}^{(s)}\big(Z_k\big|X_k=\{\boldsymbol{x}^{(s,i)}_{k|k-1}\}\big)$, $i=1,\dots,I_{\mathrm{p}}+I_{\mathrm{b}}$;
Calculate $\hat{q}^{(s)}_{k|k}$ according to (7), i.e.,
    $\hat{q}^{(s)}_{k|k} = 1 - \dfrac{\hat{f}^{(s)}(Z_k|X_k=\emptyset)\,\big(1-\hat{q}^{(s)}_{k|k-1}\big)}{\hat{f}^{(s)}(Z_k|Z_{1:k-1})}\,,$
    where (cf. (8))
    $\hat{f}^{(s)}(Z_k|Z_{1:k-1}) = C^{(s)}(Z_{1:k})\,\hat{q}^{(s)}_{k|k-1} + \hat{f}^{(s)}(Z_k|X_k=\emptyset) \times \big(1-\hat{q}^{(s)}_{k|k-1}\big)$
    with (cf. (9))
    $C^{(s)}(Z_{1:k}) = \sum_{i=1}^{I_{\mathrm{p}}+I_{\mathrm{b}}} \hat{f}^{(s)}\big(Z_k\big|X_k=\{\boldsymbol{x}^{(s,i)}_{k|k-1}\}\big)\, w^{(s,i)}_{k|k-1}\,;$
for $i=1$ to $I_{\mathrm{p}}+I_{\mathrm{b}}$ do
    Calculate a nonnormalized weight $\tilde{w}^{(s,i)}_k$ according to (10), i.e.,
        $\tilde{w}^{(s,i)}_k = \hat{f}^{(s)}\big(Z_k\big|X_k=\{\boldsymbol{x}^{(s,i)}_{k|k-1}\}\big)\, w^{(s,i)}_{k|k-1}\,;$
end for
Retain the $I_{\mathrm{p}}$ particles $\boldsymbol{x}^{(s,i)}_{k|k-1}$ with the highest weights $\tilde{w}^{(s,i)}_k$ and normalize these weights to obtain (after reindexing) a set of particles and normalized weights $\big\{\big(\boldsymbol{x}^{(s,i)}_k, w^{(s,i)}_k\big)\big\}_{i=1}^{I_{\mathrm{p}}}$;
Perform resampling if the effective sample size [12] is below a given threshold;
Output: $\hat{q}^{(s)}_{k|k}$ and $\big\{\big(\boldsymbol{x}^{(s,i)}_k, w^{(s,i)}_k\big)\big\}_{i=1}^{I_{\mathrm{p}}}$
```

VI. Simulation Results

Next, we report the results of simulations demonstrating the performance of the proposed distributed BF in a challenging scenario. For comparison, we also consider the random exchange distributed BF [7] and the centralized BF. (We note that we do not consider the consensus BF [5] because our simulation produced large errors and a divergent behavior. This can be explained by the strong nonlinearity of our measurement models combined with the fact that the consensus BF uses a linearization of the measurement model.)

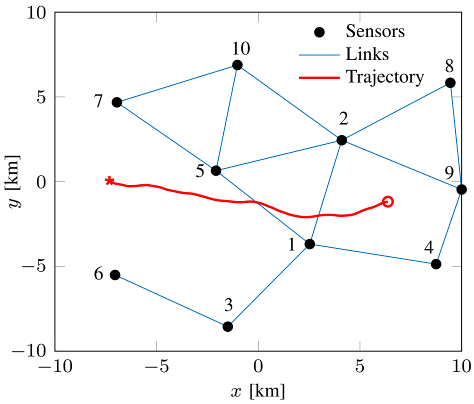

Fig. 1. Surveillance region with the sensor network and the target trajectory. The starting point of the trajectory is indicated by a star.

A. Simulation Setting

We simulated a sensor network consisting of ten sensors located in the 2-D surveillance region $[-10^4\,\mathrm{m}, 10^4\,\mathrm{m}] \times [-10^4\,\mathrm{m}, 10^4\,\mathrm{m}]$. The state vector is given by $\boldsymbol{x}_k \triangleq (x_k\;y_k\;\dot{x}_k\;\dot{y}_k)^{\mathrm{T}}$, where x_k and y_k are the target positions and $\dot{x}_k$ and $\dot{y}_k$ are the target velocities in the two coordinate directions. Fig. 1 shows the surveillance region, the sensor positions, the intersensor communication links, and the target trajectory used in our simulations. We consider time scans up to $k=120$. The target appears at time $k=10$, traverses the surveillance region from left to right, and disappears at time $k=111$.

The evolution of the state vector, when the target is present, is modeled by the near constant velocity model $\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{G}\boldsymbol{n}_k$. Here, $\boldsymbol{F} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{G} \in \mathbb{R}^{4\times 2}$ are chosen as in [18, Sec. 6.3.2] (with scan duration $T = 40\,\mathrm{s}$), and the 2-D system process $\boldsymbol{n}_k$ is modeled as i.i.d. zero-mean Gaussian with covariance matrix $\sigma_n^2\mathbf{I}_2$, where $\sigma_n = 10^{-2}\,\mathrm{m/s^2}$. Thus, $f(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}\big(\boldsymbol{x}_k; \boldsymbol{F}\boldsymbol{x}_{k-1}, \sigma_n^2\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}}\big)$. The initial prior target presence probability is $q_0 = 0.5$, i.e., we do not assume any initial information about the presence of the target. The initial prior vector pdf $f(\boldsymbol{x}_0)$ is defined implicitly by the following sampling procedure. First, the position (x_0, y_0) is drawn uniformly from the surveillance region. Then, a speed value $\nu_0 \geq 0$ is drawn from a truncated (at zero) Gaussian distribution with mean 2 and standard deviation $1/3$, and a heading angle θ_0 is drawn uniformly from the interval $[-180°, 180°)$. Finally, the velocity components are calculated as $\dot{x}_0 = \nu_0 \cos\theta_0$ and $\dot{y}_0 = \nu_0 \sin\theta_0$. Thus, we do not assume prior information about the initial position and direction of the target, but we assume that the distribution of the initial target speed is known. The target birth pdf $f_{\mathrm{b}}(\boldsymbol{x}_k)$ is chosen identical to the initial vector pdf $f(\boldsymbol{x}_0)$. The survival probability is $P_{\mathrm{s}} = 0.999$, and the birth probability $P_{\mathrm{b}} = 10^{-1}$.

We consider a heterogeneous sensor network where sensors $s \in \mathcal{S}_{\mathrm{r}} \triangleq \{2,3,4,7,9\}$ acquire range measurements

$$z^{(s)}_k = \sqrt{(x_k - x^{(s)})^2 + (y_k - y^{(s)})^2} + w^{(s)}_k\,, \qquad (21)$$

TABLE I
SENSOR PARAMETERS

s	$\sigma_{\mathrm{r}}^{(s)}$	$\sigma_{\mathrm{b}}^{(s)}$	$P_{\mathrm{d}}^{(s)}$	$\mu^{(s)}$
1	–	2°	0.8	2
2	100 m	–	0.95	5
3	150 m	–	0.9	3
4	200 m	–	0.85	2
5	–	0.5°	0.95	5
6	–	1°	0.9	3
7	100 m	–	0.95	5
8	–	2°	0.8	2
9	200 m	–	0.8	2
10	–	1°	0.9	3

and sensors $s \in \mathcal{S}_{\mathrm{b}} \triangleq \{1, 5, 6, 8, 10\}$ acquire bearing measurements

$$z_k^{(s)} = \tan^{-1}\!\left(\frac{y_k - y^{(s)}}{x_k - x^{(s)}}\right) + w_k^{(s)}. \tag{22}$$

Here, $w_k^{(s)}$ is an i.i.d. zero-mean Gaussian random process with sensor-dependent standard deviation $\sigma_{\mathrm{r}}^{(s)}$ and $\sigma_{\mathrm{b}}^{(s)}$, respectively. At each sensor s, when the target is present, target-originated measurements $z_k^{(s)}$ are randomly generated with a sensor-dependent detection probability $P_{\mathrm{d}}^{(s)}(\boldsymbol{x}_k) \equiv P_{\mathrm{d}}^{(s)}$ according to (21) or (22). In addition, clutter measurements are generated by converting the Cartesian coordinates of points uniformly sampled in the surveillance region to range or bearing values, depending on the type of sensor. In the filter algorithms, the following uniform clutter pdfs are assumed: for $s \in \mathcal{S}_{\mathrm{r}}$, $f_{\mathrm{c}}^{(s)}(z) = 1/R$ for $z \leq R$ and $f_{\mathrm{c}}^{(s)}(z) = 0$ otherwise, where $R = 30\,\mathrm{km}$ is the maximum range of the sensors; and for $s \in \mathcal{S}_{\mathrm{b}}$, $f_{\mathrm{c}}^{(s)}(z) = 1/360°$ for $z \in [-180°, 180°)$ and $f_{\mathrm{c}}^{(s)}(z) = 0$ otherwise. The number of clutter measurements is Poisson distributed with sensor-dependent mean $\mu^{(s)}$. The values of the sensor parameters $\sigma_{\mathrm{r}}^{(s)}$, $\sigma_{\mathrm{b}}^{(s)}$, $P_{\mathrm{d}}^{(s)}$, and $\mu^{(s)}$ for the various sensors are listed in Table I.

The proposed distributed BF employs a 2-D Fourier dictionary as defined in Section IV-B1 with $d = 2$ and $D_\ell = 20\,\mathrm{km}$ for $\ell = 1, 2$ (note that 20 km is the width of the surveillance area in each coordinate direction). We use a 2-D, rather than a 4-D, dictionary because the sensors produce range or bearing measurements and thus the LLFs depend only on the x_k and y_k components of the state $\boldsymbol{x}_k$. The dictionary size parameters $B'_\ell \equiv B'$ are chosen as 10, 15, or 20, so that the dictionary size $B = (2B' + 1)^2$ is 441, 961, or 1681. The LC uses Metropolis weights $\gamma_{s,s'}$ [15] and performs $J = 25$ consensus iterations. The pruning thresholds are chosen as $\eta_j = 0$ for $j = 0, \ldots, 15$ and $\eta_j = 0.01$ for $j = 16, \ldots, J$; thus, pruning sets in only after 16 consensus iterations. We use $I_{\mathrm{p}} = 1000$ predicted particles and $I_{\mathrm{b}} = 5000$ birth particles.

For comparison, we also simulated a variant of the random exchange distributed BF proposed in [7] that was adapted to our range-only or bearing-only measurement model by using the appropriate type of LLF. Following the recommendation given in [7], we use a single Gaussian for approximating the posterior distributions. The other parameters equal those defined above (if applicable). Finally, as a performance benchmark, we simulated a particle implementation of the centralized multisensor BF according to Section III.

B. Results

Fig. 2 shows three performance metrics for the proposed distributed BF with dictionary size parameter B' equal to 10, 15, or 20 (abbreviated LC-BF-10 etc.) versus time k. For each dictionary size, we show the results obtained with and without pruning. We also show the results of the random exchange BF and of the centralized BF (abbreviated RE-BF and C-BF, respectively). All results were averaged over 100 simulation runs, and those for the distributed BFs also over the ten sensors.

The root-mean-square error (RMSE) of the position estimates is shown in Fig. 2(a). For times where the target is present, i.e., $k = 10, \ldots, 110$, the RMSE was calculated at each time k by averaging over all simulation runs with successful target detections. For times where the target is absent, the RMSE was set to zero. One can see that after the appearance of the target at time $k = 10$, the RMSE of LC-BF-10, LC-BF-15, LC-BF-20, and C-BF quickly decreases to an error floor, whereas RE-BF requires about 15 time steps to reach an error floor. The peak RMSE value of RE-BF at $k = 10$ (clipped in Fig. 2(a)) is as large as about 3.3 km. LC-BF-10, LC-BF-15, and LC-BF-20 perform significantly better than RE-BF. Larger dictionary sizes yield better results: LC-BF-20 performs quite close to C-BF, and in turn LC-BF-15 performs quite close to LC-BF-20. One can also see that the pruning does not result in a significant increase in the RMSE.

Fig. 2(b) shows the estimated probability of a detection error (EPDE), which is defined as the average of an indicator variable that is one if the tracking method does not detect the target even though it is present or it detects the target even though it is absent, and zero otherwise. For all filters, a peak occurs at time $k = 10$, i.e., when the target is born. This is because the target has been observed only at this single time instant (if at all), which makes a decision about its presence unreliable. When the target is absent ($k = 1, \ldots, 9$ and $k = 111, \ldots, 120$), the EPDE of LC-BF-10/15/20 is at most about 4% and typically significantly less, and when the target is present, it is zero most of the time and in any case less than about 2%. The EPDE of RE-BF is effectively zero when the target is absent, but it has a much larger initial peak when the target is born and afterwards takes about 10 time steps until it settles to a floor of about 2%. This shows that already LC-BF-10 has a significantly better detection performance than RE-BF. Again, pruning does not have a noticeable effect on the EPDE.

Finally, Fig. 2(c) depicts the amount of communication required by LC-BF-10/15/20 —specifically, the average number of internal states (real values) that are broadcast by one sensor during the $J = 25$ consensus iterations, i.e., during one filtering step. Both with and without pruning, a larger dictionary leads to a higher amount of communication. However, as we recall from Figs. 2(a) and 2(b), a larger dictionary also yields a better detection/tracking performance. Thus, as expected, there is a tradeoff between communication cost and detection/tracking performance. Pruning is seen to yield a reduction of the communication cost by about 10–20% without noticeably affecting the detection/tracking performance. This reduction is larger—

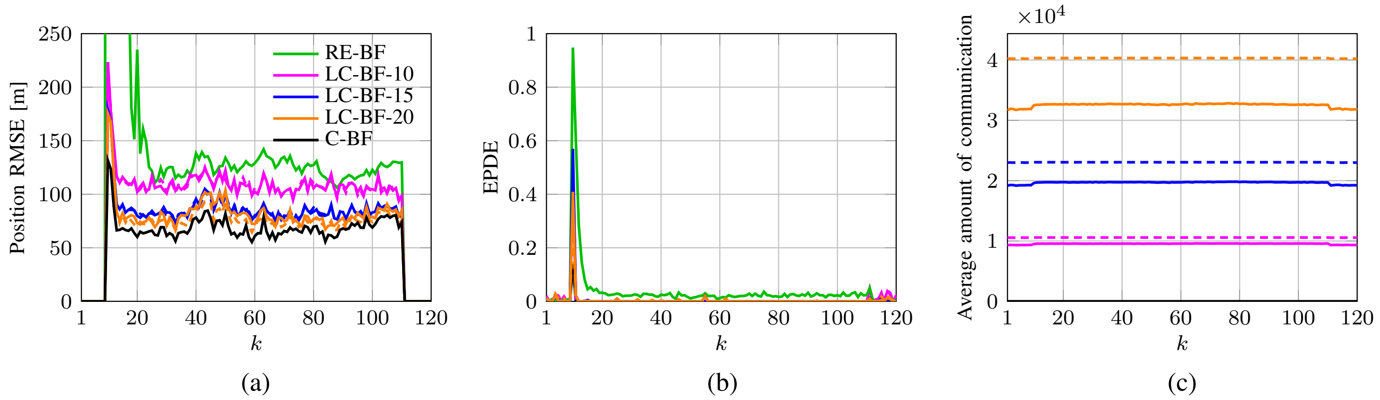

Fig. 2. Performance metrics of LC-BF-10/15/20, RE-BF, and C-BF versus time: (a) Position RMSE, (b) EPDE, (c) average number of internal states broadcast by one sensor during one filtering step. The results of LC-BF-10/15/20 with and without pruning are depicted by solid and dashed lines, respectively.

both absolutely and relatively—for a larger dictionary size. The communication cost of RE-BF is much lower than that of the proposed method. Indeed, RE-BF broadcasts only about 32 real numbers per filtering step, whereas LC-BF-10 (i.e., using the smallest of the three dictionary sizes considered) with pruning broadcasts about 10,000 real numbers per filtering step.

These results show that the proposed distributed BF achieves a significantly better performance than RE-BF, which can be very close to that of C-BF. However, this comes at the expense of a significantly larger amount of communication.

VII. Conclusion

We proposed a distributed multisensor Bernoulli filter (BF) method for target tracking in decentralized sensor networks. This method addresses the practically relevant case where the presence of the target is uncertain, the measurements are affected by clutter and missed detections, and the system model is nonlinear and non-Gaussian. In contrast to state-of-the-art distributed BFs, our method approximates the optimal Bayesian multisensor filter. It is based on an extended form of the likelihood consensus (LC) scheme that accounts for both target presence and absence and includes an adaptive pruning of the LC expansion coefficients.

Simulation results for a heterogeneous sensor network with substantial measurement noise and clutter demonstrated the excellent performance of the proposed method. For increasing size of the LC dictionary, the performance approaches that of the optimal centralized multisensor BF. The proposed method outperforms the distributed BF of [7] at the expense of a significantly higher amount of communication. Finally, the proposed pruning of the LC expansion coefficients results in a substantial reduction of communication without compromising the performance.

The proposed LC-based distributed BF is suited for the tracking of a single target. An extension to the tracking of multiple targets [10], [11], [18], [19] is an interesting direction for further research.

Acknowledgment

The authors would like to thank Dr. Pavel Rajmic for stimulating discussions.

References

[1] B. Ristic, B.-T. Vo, B.-N. Vo, and A. Farina, “A tutorial on Bernoulli filters: Theory, implementation and applications,” *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3406–3430, Jul. 2013.

[2] B. T. Vo, C. M. See, N. Ma, and W. T. Ng, “Multi-sensor joint detection and tracking with the Bernoulli filter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 2, pp. 1385–1402, Apr. 2012.

[3] G. Papa, P. Braca, S. Horn, S. Marano, V. Matta, and P. Willett, “Multisensor adaptive Bayesian tracking under time-varying target detection probability,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 5, pp. 2193–2209, Oct. 2016.

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

[5] M. B. Guldogan, “Consensus Bernoulli filter for distributed detection and tracking using multi-static Doppler shifts,” *IEEE Signal Process. Lett.*, vol. 21, no. 6, pp. 672–676, Jun. 2014.

[6] D. Clark, S. Julier, R. P. S. Mahler, and B. Ristić, “Robust multi-object sensor fusion with unknown correlations,” in *Proc. SSPD-10*, London, UK, Sep. 2010.

[7] S. S. Dias and M. G. S. Bruno, “Distributed Bernoulli filters for joint detection and tracking in sensor networks,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 3, pp. 260–275, Sep. 2016.

[8] 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.

[9] 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, Jun. 2014.

[10] R. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Norwood, MA: Artech House, 2014.

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

[12] 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.

[13] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York, NY: Springer, 1994.

[14] 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.

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

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

[17] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia, PA: SIAM, 1995.

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

[19] 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.