Reducing Communication in Distributed Bayesian Target Tracking: Likelihood Consensus 2.0

Erik Šauša, Pavel Rajmic, and Franz Hlawatsch

***Abstract*—The likelihood consensus (LC) enables a distributed computation of the global likelihood function in a decentralized sensor network with possibly nonlinear and non-Gaussian sensor characteristics. A major application of the LC is Bayes-optimal target tracking using a distributed particle filter. Here, we propose an evolved LC methodology—dubbed "LC 2.0"—with significantly reduced intersensor communication. LC 2.0 uses multiple refinements of the original LC including a sparsity-promoting calculation of expansion coefficients, the use of a B-spline dictionary, a distributed adaptive calculation of the relevant state-space region, and efficient binary representations. We consider the use of the proposed LC 2.0 within a distributed particle filter and within a distributed particle-based probabilistic data association filter. Our simulation results demonstrate that large savings in intersensor communication can be obtained without compromising the tracking performance.**

***Index Terms*—Target tracking, particle filter, likelihood consensus, splines, orthogonal matching pursuit, OMP, sparsity, PDA filter.**

I. Introduction

A. Background and Motivation

Target tracking aims at estimating the time-varying state—e.g., position and velocity—of a moving object ("target") [1], [2]. Here, we consider distributed Bayesian target tracking in a decentralized sensor network [3], based on a generally nonlinear and non-Gaussian state-space model. For distributed Bayesian target tracking, we use a distributed particle filter (DPF) combined with the likelihood consensus (LC) scheme for networkwide information dissemination [4], [5]. In addition, we consider a distributed particle-based probabilistic data association filter (PDAF), which is suited for scenarios with missed detections and clutter [2], [6]–[8].

DPF methods have been proposed and studied, e.g., in [4], [5], [9]–[17]. Besides DPF methods using the LC or other consensus-based information dissemination strategies [4], [5], [10], [16], a second class of DPF methods is based on the diffusion strategy [11]–[13], [18]. Diffusion-based DPF methods perform only one diffusion iteration per filtering time step, whereas LC-based methods perform several consensus iterations per filtering time step. However, since diffusion-based methods also exchange measurements between neighboring sensors, the communication cost can still be high. Furthermore, the diffusion approach does not aim at approximating the Bayes-optimal filter. In fact, approaching the Bayes-optimal filter would again necessitate multiple diffusion iterations per filtering time step [18].

LC-based DPF methods approximate the globally Bayes-optimal state estimator, where "globally" means that the measurements of all the sensors in the entire sensor network are taken into account. Because the globally Bayes-optimal state estimator involves the global likelihood function, the LC scheme computes an approximation thereof in a distributed way. This is achieved by first performing a dictionary expansion of the local log-likelihood function of each sensor and then disseminating and fusing the expansion coefficients by means of a consensus or gossip algorithm [5]. The communication cost of the LC increases with the accuracy of approximating the global likelihood function.

B. Contributions and Paper Organization

In this work, we propose an evolved LC methodology—dubbed "LC 2.0"—with significantly reduced communication cost. More specifically, we introduce the following modifications and extensions of the basic LC scheme:

- A "sparsity-promoting" computation of the LC expansion coefficients by means of the orthogonal matching pursuit (OMP) [19], [20]. Compared to the least-squares fit used so far, the OMP offers an improved tradeoff between approximation accuracy and communication cost by enabling an easy specification and a reduction of the number of significant expansion coefficients. We note that the OMP-based computation was previously described in our conference publication [8].
- Use of a B-spline dictionary [21], [22] instead of the Fourier or monomial dictionary used previously [4], [5], [8], [23]. The atoms of B-spline dictionaries are localized in the state space, which is advantageous in view of the locality of the posterior distribution. This entails a further modification of the LC, in which the dictionary expansion is based on the values of the local log-likelihood functions taken on a uniform grid rather than at the positions of the particles.
- An "adaptive zooming" mode of the LC in which the dictionary expansion of each local log-likelihood function is restricted to a "region of interest." This region of interest is determined online in a distributed manner.
- Efficient binary representations of the expansion coefficients communicated between the sensors.

The "LC 2.0" method proposed in this article employs an appropriate combination of OMP-based coefficient calculation,

This work was supported in part by the Austrian Science Fund (FWF) under grant P 32055-N31 and by the OeAD under WTZ grant CZ 09/2019. Parts of this paper were previously presented at the 2018 IEEE Statistical Signal Processing Workshop, Freiburg, Germany, June 2018. E. Šauša and F. Hlawatsch are with the Institute of Telecommunications, TU Wien, Vienna, Austria (e-mail: e1525267@student.tuwien.ac.at, franz.hlawatsch@tuwien.ac.at). P. Rajmic is with the Department of Telecommunications, Brno University of Technology, Brno, Czech Republic (e-mail: pavel.rajmic@vut.cz).

B-spline dictionary, uniform-grid evaluation of the local log-likelihood functions, adaptive zooming, and efficient binary representation of the expansion coefficients.

Besides the DPF, which is the simplest use case for the proposed LC 2.0, we also consider a distributed version of the particle-based PDAF described in [2], [7]. The PDAF yields an improved tracking performance in the presence of clutter and missed detections.

This article is organized as follows. Section II describes our system model and reviews LC-based distributed particle filtering. Section III presents an OMP-based calculation of the expansion coefficients and a uniform-grid evaluation of the local log-likelihood functions. In Section IV, the B-spline dictionary is introduced. A distributed method for adaptively determining a region of interest is described in Section V. Section VI presents efficient binary coefficient representations. In Section VII, we discuss the use of LC 2.0 within a distributed PDAF. Finally, the advantages of LC 2.0 are demonstrated via simulation results in Section VIII.

II. Review of LC-Based Distributed Particle Filtering

First, we review the conventional formulation of LC-based distributed particle filtering [4], [5]. The underlying system model will be extended in Section VII.

A. System Model

We consider a target with an unknown time-varying state $\mathbf{x}_n = (x_{n,1} \cdots x_{n,M})^{\mathrm{T}} \in \mathbb{R}^M$, where $n \in \mathbb{N}_0$ is a discrete time index. In many applications, the target state includes the target's position and velocity. The target state evolves according to a known state-transition probability density function (pdf) $f(\mathbf{x}_n|\mathbf{x}_{n-1})$. There are S sensors indexed by $s \in \{1,\dots,S\}$. Sensor s is able to communicate with a certain set $\mathcal{N}_s \subseteq \{1,\dots,S\} \setminus \{s\}$ of "neighboring" sensors. The graph constituted by the sensors and the communication links is assumed to be connected, i.e., there is a connection—possibly comprising multiple links—between any two sensors. At each time n, each sensor acquires a measurement such as, e.g., noisy observations of the target's range and bearing. The measurement $\mathbf{z}_n^{(s)} \in \mathbb{R}^{N_s}$ at sensor s and time n is statistically related to the target state $\mathbf{x}_n$ according to the known *local likelihood function* (LLF) $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$.

The *global likelihood function* (GLF) $f(\mathbf{z}_n|\mathbf{x}_n)$ involves $\mathbf{z}_n \triangleq \big(\mathbf{z}_n^{(1)\mathrm{T}} \cdots \mathbf{z}_n^{(S)\mathrm{T}}\big)^{\mathrm{T}}$, i.e., the measurements of all sensors at time n. We assume that the sensor measurements $\mathbf{z}_n^{(s)}$ are conditionally independent across s and n given the state sequence. It follows that the GLF factorizes into the LLFs, i.e.,

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

Each sensor s knows its own measurement $\mathbf{z}_n^{(s)}$ and its own LLF $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ (as a function of $\mathbf{x}_n$), but it does not known the measurements or LLFs of the other sensors. We emphasize that the above system model does not make any assumptions of linearity or Gaussianity.

At each time n, each sensor s estimates the current target state $\mathbf{x}_n$ from the measurements of all sensors up to time n, $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^{\mathrm{T}} \cdots \mathbf{z}_n^{\mathrm{T}})^{\mathrm{T}}$. This is achieved by an approximate distributed implementation of the minimum mean-square error (MMSE) estimator [24]

$$\hat{\mathbf{x}}_n^{\mathrm{MMSE}} \triangleq \mathrm{E}\{\mathbf{x}_n|\mathbf{z}_{1:n}\} = \int_{\mathbb{R}^M} \mathbf{x}_n\, f(\mathbf{x}_n|\mathbf{z}_{1:n})\,\mathrm{d}\mathbf{x}_n\,, \tag{2}$$

in which $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is the global posterior pdf.

B. Local Particle Filter

Each sensor s runs a *local particle filter*, which operates independently of the other sensors except that it uses an approximation of the GLF $f(\mathbf{z}_n|\mathbf{x}_n)$ that is calculated in a distributed way via the LC scheme (see Section II-C). At each time n, the local particle filter at sensor s represents the global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ by a set of J particles and associated weights, $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^{J}$, with $\sum_{j=1}^{J} w_n^{(s,j)} = 1$.

Using the simplest particle filter algorithm [25], this particle representation is calculated time-recursively as follows. In the *prediction step*, for each previous particle $\mathbf{x}_{n-1}^{(s,j)}$, a "predicted" particle $\mathbf{x}_{n|n-1}^{(s,j)}$ 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 $\mathbf{x}_{n-1} = \mathbf{x}_{n-1}^{(s,j)}$. In the *update step*, the associated weights are calculated as

$$w_{n|n-1}^{(s,j)} = c\,\hat{f}_s\big(\mathbf{z}_n\big|\mathbf{x}_{n|n-1}^{(s,j)}\big), \quad j = 1,\dots,J, \tag{3}$$

with normalization factor $c = 1/\sum_{j=1}^{J} \hat{f}_s\big(\mathbf{z}_n\big|\mathbf{x}_{n|n-1}^{(s,j)}\big)$. Here, $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ denotes an approximation to the GLF $f(\mathbf{z}_n|\mathbf{x}_n)$ in (1) that involves the current measurements of all the sensors, $\mathbf{z}_n$. This GLF approximation is calculated in a distributed way via the LC scheme reviewed in Section II-C, which requires communication with all the neighboring sensors $s' \in \mathcal{N}_s$. Next, the weighted particle set $\big\{\big(\mathbf{x}_{n|n-1}^{(s,j)}, w_{n|n-1}^{(s,j)}\big)\big\}_{j=1}^{J}$ is *resampled* to avoid particle degeneracy [25], [26]; this results in the new particles $\mathbf{x}_n^{(s,j)}$, $j = 1,\dots,J$ with associated weights $w_n^{(s,j)} \equiv 1/J$. The overall recursive algorithm is initialized at time $n = 0$ by particles $\mathbf{x}_0^{(s,j)}$, $j = 1,\dots,J$ that are randomly drawn from some prior pdf $f(\mathbf{x}_0)$, and by the weights $w_0^{(s,j)} \equiv 1/J$. Finally, the local particle filter at sensor s calculates an approximation $\hat{\mathbf{x}}_n^{(s)}$ to the MMSE estimate $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$ in (2) as the weighted sample mean of the predicted particles (before resampling), i.e, $\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^{J} w_{n|n-1}^{(s,j)} \mathbf{x}_{n|n-1}^{(s,j)}$.

C. The LC Scheme

Next, we review the LC scheme [4], [5], which is used for the distributed calculation of the GLF approximations $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ involved in the update step (3). Let us consider

$$L_n(\mathbf{x}_n) \triangleq \frac{1}{S}\log f(\mathbf{z}_n|\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S} \log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big), \tag{4}$$

where $\log$ denotes the natural logarithm and (1) was used. Note that, conversely, $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(S L_n(\mathbf{x}_n))$. Using a dictionary of functions or "atoms" $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ that is identical for all sensors, each sensor s approximates its log-LLF by a linear combination of the atoms, i.e.,

$$\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) \approx \sum_{k=1}^{K} \alpha_n^{(s,k)} \psi_k(\mathbf{x}_n). \tag{5}$$

Here, the local expansion coefficients $\big\{\alpha_n^{(s,k)}\big\}_{k=1}^{K}$ are calculated locally at each sensor s using the local measurements $\mathbf{z}_n^{(s)}$, as described in Section III. The choice of the dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ will be considered in Section IV.

By combining (5) with (4), we easily obtain the following approximation of $L_n(\mathbf{x}_n) = \frac{1}{S}\log f(\mathbf{z}_n|\mathbf{x}_n)$:

$$L_n(\mathbf{x}_n) \approx \sum_{k=1}^{K} \beta_n^{(k)} \psi_k(\mathbf{x}_n), \tag{6}$$

with the *global* expansion coefficients $\beta_n^{(k)} \triangleq \frac{1}{S}\sum_{s=1}^{S} \alpha_n^{(s,k)}$, $k = 1, \ldots, K$. As the global expansion coefficients $\beta_n^{(k)}$ are the averages of the associated local expansion coefficients $\alpha_n^{(s,k)}$ of the individual sensors, they can be computed in a distributed manner by means of K instances of the average consensus algorithm [27], which merely requires each sensor s to communicate with its neighboring sensors $s' \in \mathcal{N}_s$. In iteration $i \in \{1, 2, \ldots\}$ of the kth instance of the average consensus algorithm, sensor s updates an iterated estimate of $\beta_n^{(k)}$ as

$$\hat{\beta}_n^{(k,s)}[i] = \sum_{s' \in \{s\} \cup \mathcal{N}_s} \gamma_{s,s'} \hat{\beta}_n^{(k,s')}[i-1]. \tag{7}$$

Here, the $\gamma_{s,s'}$ are suitably chosen weights [27]–[29]; a simple standard choice is given by the Metropolis weights [28], [29]. Furthermore, the $\hat{\beta}_n^{(k,s')}[i-1]$ for all $s' \in \mathcal{N}_s$ were communicated to sensor s by its neighboring sensors $s' \in \mathcal{N}_s$. Sensor s then broadcasts the updated iterated coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$, $k = 1, \ldots, K$ to its neighboring sensors.

The recursion (7) is initialized by the local expansion coefficient, i.e., $\hat{\beta}_n^{(k,s)}[0] = \alpha_n^{(s,k)}$, and terminated after a sufficient number I of iterations. The final estimates $\hat{\beta}_n^{(k,s)}[I]$ are then used in (6) to obtain an approximation to $L_n(\mathbf{x}_n)$ and, in turn, to the GLF $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(SL_n(\mathbf{x}_n))$. Thus, the approximation to $f(\mathbf{z}_n|\mathbf{x}_n)$ obtained at sensor s is

$$\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n) = \exp\left(S \sum_{k=1}^{K} \hat{\beta}_n^{(k,s)}[I]\, \psi_k\big(\mathbf{x}_n\big)\right).$$

For $I \to \infty$, the consensus recursion would converge to $\beta_n^{(k)}$ because, as assumed in Section II-A, the communication graph is connected [28]. As an alternative to the average consensus algorithm, a gossip algorithm [30] can be used.

In each iteration i of the average consensus algorithm, sensor s has to broadcast the real numbers $\hat{\beta}_n^{(k,s)}[i]$, $k = 1, \ldots, K$ to its neighboring sensors $s' \in \mathcal{N}_s$. In the next four sections, we will present modifications of the original LC scheme that lead to a significant reduction of the communication cost.

III. Sparsity-promoting LLF Approximation using the OMP

In this section, we propose a sparsity-promoting method for calculating the local expansion coefficients $\big\{\alpha_n^{(s,k)}\big\}_{k=1}^{K}$ that are involved in the log-LLF approximation (5) for a given number K of atoms $\psi_k(\mathbf{x})$.

A. Review of Least-Squares-based LLF Approximation

The method originally proposed in [4], [5] performs a least squares (LS) fit of the right hand side of (5) to the left hand side in a way such that the total approximation error at the predicted particles $\big\{\mathbf{x}_{n|n-1}^{(s,j)}\big\}_{j=1}^{J}$ is minimized. Let us introduce the local coefficient vector $\boldsymbol{\alpha}_n^{(s)}$, the discretized-atom vector $\boldsymbol{\psi}_{n,k}^{(s)}$, and the discretized-log-LLF vector $\boldsymbol{\eta}_n^{(s)}$ as

$$\begin{aligned}
\boldsymbol{\alpha}_n^{(s)} &\triangleq \big(\alpha_n^{(s,1)} \cdots \alpha_n^{(s,K)}\big)^{\mathrm{T}} \in \mathbb{R}^K,\\
\boldsymbol{\psi}_{n,k}^{(s)} &\triangleq \big(\psi_k(\mathbf{x}_{n|n-1}^{(s,1)}) \cdots \psi_k(\mathbf{x}_{n|n-1}^{(s,J)})\big)^{\mathrm{T}} \in \mathbb{R}^J,\\
\boldsymbol{\eta}_n^{(s)} &\triangleq \big(\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_{n|n-1}^{(s,1)}) \cdots \log f(\mathbf{z}_n^{(s)}|\mathbf{x}_{n|n-1}^{(s,J)})\big)^{\mathrm{T}} \in \mathbb{R}^J.
\end{aligned}$$

Then, the error minimized by the LS fit is given by

$$\epsilon_n^{(s)} = \left\| \boldsymbol{\eta}_n^{(s)} - \sum_{k=1}^{K} \alpha_n^{(s,k)} \boldsymbol{\psi}_{n,k}^{(s)} \right\|^2 = \big\| \boldsymbol{\eta}_n^{(s)} - \boldsymbol{\Psi}_n^{(s)} \boldsymbol{\alpha}_n^{(s)} \big\|^2,$$

where the discretized-dictionary matrix $\boldsymbol{\Psi}_n^{(s)} \in \mathbb{R}^{J \times K}$ has the vectors $\boldsymbol{\psi}_{n,k}^{(s)}$, $k = 1, \ldots, K$ as columns. Note that $\boldsymbol{\psi}_{n,k}^{(s)}$ and $\boldsymbol{\Psi}_n^{(s)}$ depend on the predicted particles $\mathbf{x}_{n|n-1}^{(s,j)}$. The coefficient vector $\boldsymbol{\alpha}_n^{(s)}$ minimizing $\epsilon_n^{(s)}$ is given by [31]

$$\boldsymbol{\alpha}_{n,\mathrm{LS}}^{(s)} = \boldsymbol{\Psi}_n^{(s)+} \boldsymbol{\eta}_n^{(s)},$$

where $\boldsymbol{\Psi}_n^{(s)+} \triangleq \big(\boldsymbol{\Psi}_n^{(s)\mathrm{T}} \boldsymbol{\Psi}_n^{(s)}\big)^{-1} \boldsymbol{\Psi}_n^{(s)\mathrm{T}}$ is the Moore–Penrose pseudoinverse of $\boldsymbol{\Psi}_n^{(s)}$. Here, it is assumed that the vectors $\boldsymbol{\psi}_{n,k}^{(s)}$ are linearly independent and $J \geq K$, i.e., the number of particles is larger than or equal to the number of atoms.

B. OMP-based LLF Approximation

As an improvement over this LS-based calculation, we propose a sparsity-promoting calculation using the OMP [19], [20]. Our goal 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.

The OMP is a greedy iterative algorithm that selects one atom per iteration. In the first iteration, the OMP at sensor s selects the atom index k for which the ℓ_2-normalized atom vector $\boldsymbol{\psi}_{n,k}^{(s)}$ best matches the log-LLF vector $\boldsymbol{\eta}_n^{(s)}$, i.e.,

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

(Note that k_1 also depends on n and s, which is however not indicated for notational simplicity.) Then, the *residual* $\boldsymbol{\rho}_{n,1}^{(s)}$ is formed by subtracting from $\boldsymbol{\eta}_n^{(s)}$ the orthogonal projection of $\boldsymbol{\eta}_n^{(s)}$ onto $\boldsymbol{\psi}_{n,k_1}^{(s)}$, i.e., $\boldsymbol{\rho}_{n,1}^{(s)} = \boldsymbol{\eta}_n^{(s)} - a_1 \boldsymbol{\psi}_{n,k_1}^{(s)}$ with $a_1 = \boldsymbol{\psi}_{n,k_1}^{(s)\mathrm{T}} \boldsymbol{\eta}_n^{(s)} / \|\boldsymbol{\psi}_{n,k_1}^{(s)}\|$. In the further iterations $l = 2, 3, \ldots$, the atom index k for which the normalized version of $\boldsymbol{\psi}_{n,k}^{(s)}$ best matches the previous residual $\boldsymbol{\rho}_{n,l-1}^{(s)}$ is selected, i.e.,

$$k_l = \operatorname*{argmax}_{k \in \{1,\ldots,K\}} \frac{\big|\boldsymbol{\psi}_{n,k}^{(s)\mathrm{T}} \boldsymbol{\rho}_{n,l-1}^{(s)}\big|}{\|\boldsymbol{\psi}_{n,k}^{(s)}\|}.$$

Then, a new residual $\boldsymbol{\rho}_{n,l}^{(s)}$ is formed by subtracting from $\boldsymbol{\eta}_n^{(s)}$ the orthogonal projection of $\boldsymbol{\eta}_n^{(s)}$ onto the subspace of $\mathbb{R}^J$

spanned by $\boldsymbol{\psi}_{n,k_1}^{(s)},\ldots,\boldsymbol{\psi}_{n,k_l}^{(s)}$, i.e.,

$$\boldsymbol{\rho}_{n,l}^{(s)} = \boldsymbol{\eta}_n^{(s)} - \mathbf{P}_{n,l}^{(s)}\boldsymbol{\eta}_n^{(s)}, \tag{8}$$

with the orthogonal projection matrix

$$\mathbf{P}_{n,l}^{(s)} \triangleq \boldsymbol{\Phi}_{n,l}^{(s)}\big(\boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}}\boldsymbol{\Phi}_{n,l}^{(s)}\big)^{-1}\boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}} = \boldsymbol{\Phi}_{n,l}^{(s)}\boldsymbol{\Phi}_{n,l}^{(s)+}, \tag{9}$$

where $\boldsymbol{\Phi}_{n,l}^{(s)} \in \mathbb{R}^{J\times l}$ is the matrix with columns $\boldsymbol{\psi}_{n,k_1}^{(s)},\ldots,\boldsymbol{\psi}_{n,k_l}^{(s)}$. We note that (8), (9) can be rewritten as

$$\boldsymbol{\rho}_{n,l}^{(s)} = \boldsymbol{\eta}_n^{(s)} - \boldsymbol{\Phi}_{n,l}^{(s)}\mathbf{c}_{n,l}^{(s)}, \quad \text{with } \mathbf{c}_{n,l}^{(s)} \triangleq \boldsymbol{\Phi}_{n,l}^{(s)+}\boldsymbol{\eta}_n^{(s)}.$$

Here, $\mathbf{c}_{n,l}^{(s)} \in \mathbb{R}^l$ is the LS solution to the problem of approximating $\boldsymbol{\eta}_n^{(s)}$ by $\boldsymbol{\Phi}_{n,l}^{(s)}\mathbf{c}$, i.e., $\mathbf{c}_{n,l}^{(s)} = \operatorname{argmin}_{\mathbf{c}} \big\|\boldsymbol{\eta}_n^{(s)} - \boldsymbol{\Phi}_{n,l}^{(s)}\mathbf{c}\big\|^2$.

This iterative algorithm is stopped after a specified number of iterations or when $\|\boldsymbol{\rho}_{n,l}^{(s)}\|$ falls below a specified positive threshold. Let L_s denote the final iteration index at sensor s. Then the result of the OMP algorithm is the coefficient vector $\boldsymbol{\alpha}_{n,\mathrm{OMP}}^{(s)} = \big(\alpha_{n,\mathrm{OMP}}^{(s,1)}\cdots\alpha_{n,\mathrm{OMP}}^{(s,K)}\big)^{\mathrm{T}}$ whose elements $\alpha_{n,\mathrm{OMP}}^{(s,k)}$ are equal to the associated elements of $\mathbf{c}_{n,L_s}^{(s)} = \boldsymbol{\Phi}_{n,L_s}^{(s)+}\boldsymbol{\eta}_n^{(s)} \in \mathbb{R}^{L_s}$ if $k \in \{k_1,\ldots,k_{L_s}\}$ and zero otherwise, i.e.,

$$\alpha_{n,\mathrm{OMP}}^{(s,k)} = \begin{cases} \big(\mathbf{c}_{n,L_s}^{(s)}\big)_l, & k = k_l \in \{k_1,\ldots,k_{L_s}\}, \\ 0, & k \notin \{k_1,\ldots,k_{L_s}\}. \end{cases}$$

Accordingly, only L_s of the K elements of $\boldsymbol{\alpha}_{n,\mathrm{OMP}}^{(s)}$ are nonzero, which means that the number of nonzero local expansion coefficients equals the number of OMP iterations performed. Thus, depending on the stopping criterion, $\boldsymbol{\alpha}_{n,\mathrm{OMP}}^{(s)}$ is a more or less sparse vector. The computational complexity of the OMP is higher than that of the LS-based calculation because the OMP comprises L_s LS problems involving inversion of the matrices $\boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}}\boldsymbol{\Phi}_{n,l}^{(s)} \in \mathbb{R}^{l\times l}$ for $l = 1,\ldots,L_s$. Note, however, that the matrices $\boldsymbol{\Phi}_{n,l}^{(s)}$ are typically small.

C. Uniform Sampling of the Log-LLF

In both the OMP-based and LS-based calculation of the local expansion coefficients $\big\{\alpha_n^{(s,k)}\big\}_{k=1}^K$, the log-LLF $\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ and the approximating function $\sum_{k=1}^K \alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$ are sampled at the predicted particles $\mathbf{x}_n = \mathbf{x}_{n|n-1}^{(s,j)}$, $j = 1,\ldots,J$. However, when using a B-spline dictionary as proposed in Section IV-B, we observed experimentally that the expansion coefficients are often unrealistically large, which leads to an incorrect selection of the set of significant atoms that are used for approximating the LLF. This is probably due to the fact that, as B-spline atoms are highly localized in the state space, the expansion coefficient for a B-spline atom that is located away from the main particle population can be heavily affected by a few local "outlier particles" that are not representative of the posterior pdf. This effect does not occur in the case of a Fourier dictionary, because the Fourier atoms are not localized and thus the respective coefficients are always affected by all the particles.

To avoid this issue, we propose to use a *uniform* sampling of $\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ and $\sum_{k=1}^K \alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$, where the samples $\mathbf{x}_n^{(q)} = \big(x_{n,1}^{(q)}\cdots x_{n,M}^{(q)}\big)^{\mathrm{T}}$, $q = 1,\ldots,Q_n$ lie on a regular grid within a *region of interest* (ROI)

$$\mathcal{R}_n = \big[a_n^{(1)}, b_n^{(1)}\big] \times \ldots \times \big[a_n^{(M)}, b_n^{(M)}\big] \subset \mathbb{R}^M. \tag{10}$$

More specifically, in the mth coordinate direction, there are $Q_n^{(m)}$ sample points uniformly spaced in the interval $\big[a_n^{(m)}, b_n^{(m)}\big]$. The total number of M-dimensional (M-D) samples $\mathbf{x}_n^{(q)}$ then is $Q_n = \prod_{m=1}^M Q_n^{(m)}$. A distributed, particle-based method for calculating the ROI interval bounds $a_n^{(m)}$ and $b_n^{(m)}$ will be presented in Section V.

Using this uniform sampling approach, the atom vector $\boldsymbol{\psi}_{n,k}$ and the log-LLF vector $\boldsymbol{\eta}_n^{(s)}$ are redefined as

$$\boldsymbol{\psi}_{n,k} \triangleq \big(\psi_k(\mathbf{x}_n^{(1)})\cdots\psi_k(\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}} \in \mathbb{R}^{Q_n},$$

$$\boldsymbol{\eta}_n^{(s)} \triangleq \big(\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(1)})\cdots\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}} \in \mathbb{R}^{Q_n}.$$

We also obtain new dictionary matrices $\boldsymbol{\Psi}_n \in \mathbb{R}^{Q_n\times K}$ (for LS) and $\boldsymbol{\Phi}_{n,l}^{(s)} \in \mathbb{R}^{Q_n\times l}$ (for OMP), as the $\boldsymbol{\psi}_{n,k}$ constitute the columns of these matrices. Note that $\boldsymbol{\Phi}_{n,l}^{(s)}$ still depends on s since the OMP-based selection of the $\boldsymbol{\psi}_{n,k}$ constituting the columns of $\boldsymbol{\Phi}_{n,l}^{(s)}$ depends on the LLF at sensor s.

IV. B-Spline Dictionary

The dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^K$ used in the log-LLF approximation (5) affects both the accuracy of the approximation and the communication cost of the LC. In this section, we introduce the B-spline dictionary as an attractive alternative to the Fourier dictionary used in the conventional LC [5], [8], [23]. We temporarily drop the time index n for notational simplicity.

A. Review of the Fourier Dictionary

To prepare the ground, we briefly review the Fourier dictionary, which will also be used as a benchmark in our simulations in Section VIII. We first consider 1-D Fourier dictionaries. The atoms of the 1-D Fourier dictionary for the mth coordinate direction, $\big\{\psi_{\tilde{k}}^{(m)}(x)\big\}_{\tilde{k}=1}^{2\tilde{K}_m+1}$, with $m \in \{1,\ldots,M\}$, are defined as

$$\psi_{\tilde{k}}^{(m)}(x) = \cos\left(\frac{2\pi}{d^{(m)}}(\tilde{k}-1)(x-a^{(m)})\right)$$

for $\tilde{k} = 1,\ldots,\tilde{K}_m+1$ and

$$\psi_{\tilde{k}}^{(m)}(x) = \sin\left(\frac{2\pi}{d^{(m)}}(\tilde{k}-1-\tilde{K}_m)(x-a^{(m)})\right)$$

for $\tilde{k} = \tilde{K}_m+2,\ldots,2\tilde{K}_m+1$, all for $x \in \big[a^{(m)}, b^{(m)}\big]$. Here, $d^{(m)} \triangleq b^{(m)} - a^{(m)}$ is the ROI interval length in the mth coordinate direction. Note that the number of frequencies involved (including frequency 0) is $\tilde{K}_m+1$. The overall M-D Fourier dictionary on the ROI $\mathcal{R}$ is then constructed as

$$\tilde{\psi}_{\tilde{\mathbf{k}}}(\mathbf{x}) = \prod_{m=1}^M \psi_{\tilde{k}_m}^{(m)}(x_m), \tag{11}$$

with M-D index $\tilde{\mathbf{k}} \triangleq (\tilde{k}_1\cdots\tilde{k}_M)^{\mathrm{T}} \in \{1,\ldots,2\tilde{K}_1+1\} \times \ldots \times \{1,\ldots,2\tilde{K}_M+1\}$. Finally, mapping $\tilde{\mathbf{k}}$ to a 1-D index $k \in \{1,\ldots,K\}$, with $K = \prod_{m=1}^M(2\tilde{K}_m+1)$, yields the M-D Fourier dictionary with 1-D indexing $\{\psi_k(\mathbf{x})\}_{k=1}^K$.

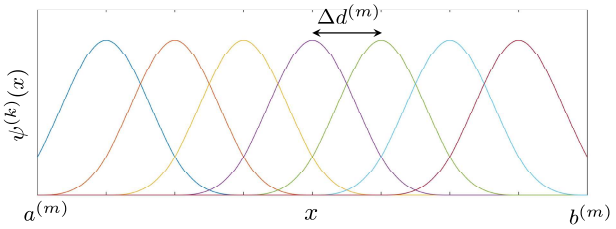


Fig. 1: 1-D B-spline dictionary as defined by (13), with $a^{(m)}=0$, $b^{(m)}=8$, $\tilde{K}_m=7$, and $\Delta d^{(m)}=(b^{(m)}-a^{(m)})/(\tilde{K}_m+1)=1$.

B. The B-Spline Dictionary

As an alternative to the Fourier dictionary, we propose the use of a dictionary of M-D B-splines with uniform spacing of their knots [21]. The advantage of B-spline dictionaries is the localization of their atoms, which is desirable in view of the localization of the posterior distribution.

A 1-D B-spline of degree $r \in \mathbb{N}_0$ is a piecewise polynomial function composed of polynomial segments of degree r. For example, the 1-D cubic (i.e., $r=3$) B-spline prototype with knots positioned at the integers is an even function with support $(-2,2)$ that is given by [22]

$$\psi(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{1}{2}|x|^3, & 0 \le |x| < 1 \\ \frac{1}{6}(2-|x|)^3, & 1 \le |x| < 2 \\ 0, & |x| \notin [0,2). \end{cases} \tag{12}$$

B-splines enjoy numerous desirable properties [22]. In particular, their atoms have compact support, and thus are localized. This is an important difference from the Fourier dictionary, whose atoms are supported on the entire ROI. Furthermore, B-splines are continuous and have continuous derivatives up to degree $r-1$. They are well suited for interpolation; asymptotically (for $r \to \infty$), they achieve the interpolation properties of the sinc function. Finally, the infinite sum of all shifts $\psi(x-i)$, $i \in \mathbb{Z}$ of a 1-D B-spline prototype $\psi(x)$ is a constant. Analogous properties hold in the M-D case.

We first consider 1-D B-spline dictionaries. The atoms of the 1-D B-spline dictionary for the mth coordinate direction, $\{\psi^{(m)}_{\tilde{k}}(x)\}_{\tilde{k}=1}^{\tilde{K}_m}$, are defined by scaling and shifting the B-spline prototype $\psi(x)$ in (12) according to

$$\psi^{(m)}_{\tilde{k}}(x) = \psi\left(\frac{x - a^{(m)} - \tilde{k}\Delta d^{(m)}}{\Delta d^{(m)}}\right), \tag{13}$$

for $\tilde{k} = 1, \ldots, \tilde{K}_m$, where $x \in \left[a^{(m)}, b^{(m)}\right]$. Here, $\Delta d^{(m)} \triangleq (b^{(m)} - a^{(m)})/(\tilde{K}_m + 1)$ is the grid spacing and $\tilde{K}_m$ is the number of shifts in the mth coordinate direction. The 1-D B-spline atoms $\psi^{(m)}_{\tilde{k}}(x)$ are centered around grid points $x_{\tilde{k}} = a^{(m)} + \tilde{k}\Delta d^{(m)}$, $\tilde{k} = 1, \ldots, \tilde{K}_m$ that are placed uniformly in the interval $\left[a^{(m)}, b^{(m)}\right]$ with spacing $\Delta d^{(m)}$. An example is shown in Fig. 1.

The M-D B-spline dictionary on the ROI $\mathcal{R}$ is then composed of M-D atoms $\tilde{\psi}_{\tilde{\mathbf{k}}}(\mathbf{x})$ that are constructed as in (11), with M-D index $\tilde{\mathbf{k}} \triangleq (\tilde{k}_1 \cdots \tilde{k}_M)^{\mathrm{T}} \in \{1, \ldots, \tilde{K}_1\} \times \ldots \times \{1, \ldots, \tilde{K}_M\}$. Finally, the M-D dictionary with 1-D indexing, $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$, is obtained by mapping $\tilde{\mathbf{k}}$ to a 1-D index $k \in \{1, \ldots, K\}$, with $K = \prod_{m=1}^{M} \tilde{K}_m$. By this construction, the M-D atoms are shifts of an M-D B-spline prototype that are located on an M-D grid. This grid is regular in each coordinate direction.

V. Distributed Calculation of the ROI

According to Section IV, the choice of the ROI $\mathcal{R}_n$ in (10) influences the number of the B-spline or Fourier atoms $\psi_k(\mathbf{x})$, and thus also the number of expansion coefficients $\alpha_n^{(s,k)}$ that are communicated between neighboring sensors. Furthermore, if the LC employs uniform sampling of $\log f\big(\mathbf{z}_n^{(s)} \big| \mathbf{x}_n\big)$ as described in Section III-C, then the choice of $\mathcal{R}_n$ also influences the number Q_n of M-D samples $\mathbf{x}_n^{(1)}, \ldots, \mathbf{x}_n^{(Q_n)}$. It follows that the choice of $\mathcal{R}_n$ has an influence on the accuracy of the log-LLF approximation as well as on the communication cost of the LC. In this section, we present a distributed algorithm that adaptively determines the ROI $\mathcal{R}_n$. The goal is to "zoom in" on the effective support of the current global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ in order to avoid sampling the log-LLF and placing atoms outside that effective support, i.e., using computation and communication resources to approximate the log-LLF on irrelevant parts of the surveillance area.

A. Calculation of the ROI Interval Bounds

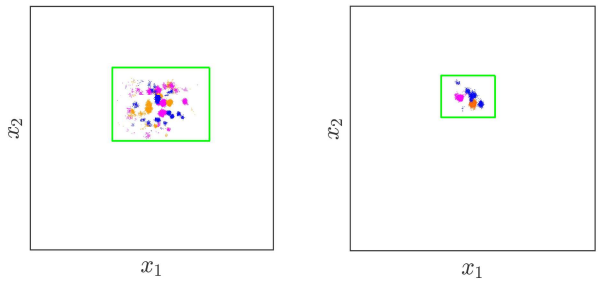
Our task is to determine the interval bounds $a_n^{(m)}$ and $b_n^{(m)}$ of the ROI $\mathcal{R}_n = \left[a_n^{(1)}, b_n^{(1)}\right] \times \ldots \times \left[a_n^{(M)}, b_n^{(M)}\right]$. It will be convenient to parametrize $\mathcal{R}_n$ in terms of the center point $\boldsymbol{\xi}_n = (\xi_n^{(1)} \cdots \xi_n^{(M)})^{\mathrm{T}}$ with $\xi_n^{(m)} \triangleq (a_n^{(m)} + b_n^{(m)})/2$ and the extension around the center point as characterized by the "extent vector" $\boldsymbol{d}_n = (d_n^{(1)} \cdots d_n^{(M)})^{\mathrm{T}}$ with $d_n^{(m)} = b_n^{(m)} - a_n^{(m)}$. The interval bounds can be recovered from $\boldsymbol{\xi}_n$ and $\boldsymbol{d}_n$ as $a_n^{(m)} = \xi_n^{(m)} - d_n^{(m)}/2$ and $b_n^{(m)} = \xi_n^{(m)} + d_n^{(m)}/2$.

The proposed algorithm for calculating the ROI determines $\boldsymbol{\xi}_n$ and $\boldsymbol{d}_n$ at each time n in a distributed manner. This has to be done before the LC is performed, i.e., before the update step of the local particle filter. Indeed, the update step presupposes knowledge of the ROI, because all sensors use the same ROI-dependent dictionary. Therefore, when the ROI is calculated, the current particle weights $w_n^{(s,j)}$ are not yet available.

First, each sensor s calculates the sample variances of the predicted particles $\{\mathbf{x}_{n|n-1}^{(s,j)}\}_{j=1}^{J}$ in the individual coordinate directions,

$$\sigma_{n,m}^{(s)2} \triangleq \frac{1}{J}\sum_{j=1}^{J}\left(x_{n|n-1,m}^{(s,j)} - \hat{x}_{n,m}^{(s)}\right)^2, \quad m = 1, \ldots, M,$$

where $\hat{x}_{n,m}^{(s)} \triangleq \frac{1}{J}\sum_{j=1}^{J} x_{n|n-1,m}^{(s,j)}$; here, $x_{n|n-1,m}^{(s,j)}$ denotes the mth element of $\mathbf{x}_{n|n-1}^{(s,j)}$. Then, for each m, $\hat{x}_{n,m}^{(s)}$ and $\sigma_{n,m}^{(s)2}$ are averaged across all sensors, i.e., the goal is to compute $\bar{\hat{x}}_{n,m} \triangleq \frac{1}{S}\sum_{s=1}^{S}\hat{x}_{n,m}^{(s)}$ and $\overline{\sigma_{n,m}^2} \triangleq \frac{1}{S}\sum_{s=1}^{S}\sigma_{n,m}^{(s)2}$. These averaging operations are implemented in a distributed manner by performing $2M$ parallel instances of the average consensus algorithm to compute approximations $\bar{\hat{x}}_{n,m}^{(s)}$ of $\bar{\hat{x}}_{n,m}$ and $\overline{\sigma_{n,m}^2}^{(s)}$ of $\overline{\sigma_{n,m}^2}$, for $m = 1, \ldots, M$. As only a finite number of consensus iterations can be performed, the approximations $\bar{\hat{x}}_{n,m}^{(s)}$ and $\overline{\sigma_{n,m}^2}^{(s)}$ obtained at different sensors s will be (slightly) different. Because the LC requires the same ROI $\mathcal{R}_n$ at each sensor,



(a) $n=3$ (b) $n=4$

Fig. 2: Example (simulation result) of the ROI $\mathcal{R}_n$ for dimension $M=2$ at two successive time points n. The ROI boundary is shown as a green rectangle. The colored dots depict the predicted particles $\mathbf{x}_{n|n-1}^{(s,j)}$ for three sensors $s=1,2,3$; the color indicates the sensor.

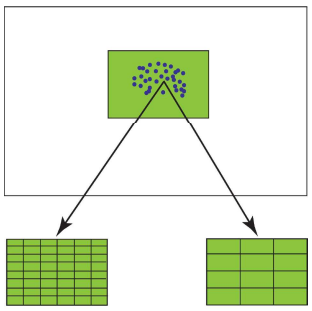


Fig. 3: Illustration of the covering of the ROI $\mathcal{R}_n$ (green rectangle) with B-spline atoms of two different densities, for dimension $M=2$. Also shown are the predicted particles $\mathbf{x}_{n|n-1}^{(s,j)}$ for all the sensors (blue dots). The small grid rectangles within the ROI indicate the effective supports of the 2-D B-spline atoms.

the next step in the algorithm is to perform $2M$ parallel instances of the maximum consensus algorithm [32] to compute $\bar{\hat{x}}_{n,m}^{\max} \triangleq \max\{\bar{\hat{x}}_{n,m}^{(s)}\}_{s=1}^{S}$ and $\overline{\sigma_{n,m}^2}^{\max} \triangleq \max\{\overline{\sigma_{n,m}^2}^{(s)}\}_{s=1}^{S}$ for $m=1,\ldots,M$. The maximum consensus algorithm converges in a finite number of iterations, which equals the diameter of the sensor network, i.e., the minimum number of hops connecting the two most distant nodes of the network [32]. Hence, after performing these iterations, identical maxima $\bar{\hat{x}}_{n,m}^{\max}$ and $\overline{\sigma_{n,m}^2}^{\max}$ are available at all sensors.

We then define the ROI center point $\boldsymbol{\xi}_n$ to be the vector with elements

$$\xi_n^{(m)} = \bar{\hat{x}}_{n,m}^{\max}, \quad m=1,\ldots,M.$$

Next, we choose each extent vector element $d_n^{(m)}$ as a function of the standard deviation $s_{n,m} \triangleq \sqrt{\overline{\sigma_{n,m}^2}^{\max}}$. The goal is to choose the $d_n^{(m)}$ sufficiently large so that $\mathcal{R}_n$ tends to include all the particles, but not larger because this would result in an unnecessarily large dictionary size K. Thus, we set $d_n^{(m)}$ equal to $\gamma\, s_{n,m}$, where $\gamma>1$ is an empirically chosen scaling factor, subject to a lower bound $d_{\min}^{(m)}$, i.e.,

$$d_n^{(m)} = \begin{cases} \gamma\, s_{n,m}, & \text{if } \gamma\, s_{n,m} \geq d_{\min}^{(m)}, \\ d_{\min}^{(m)}, & \text{if } \gamma\, s_{n,m} < d_{\min}^{(m)}. \end{cases} \tag{14}$$

Here, the lower bound $d_{\min}^{(m)}$ adds robustness in cases where the set of predicted particles $\{\mathbf{x}_{n|n-1}^{(s,j)}\}_{j=1}^{J}$ is very concentrated in the state space. Finally, if necessary, this ROI is replaced by its intersection with the overall state space region that is available for $\mathbf{x}_n$. Fig. 2 shows an example of the ROI $\mathcal{R}_n$ at two successive time points n along with the predicted particles.

B. Calculation of the Numbers of Atoms

Once the extent parameters $d_n^{(m)}$ are known, we are also able to calculate the numbers of 1-D B-spline or Fourier atoms $\psi_{\tilde{k}}^{(m)}(x)$. We recall from Section IV-B that the number and spacing of the 1-D B-spline atoms in the mth coordinate direction are given by $\tilde{K}_{n,m}$ and $\Delta d_n^{(m)} = d_n^{(m)}/(\tilde{K}_{n,m}+1)$, respectively. Therefore, for a specified B-spline spacing $\Delta d_n^{(m)}$, we obtain

$$\tilde{K}_{n,m} = \left\lceil \frac{d_n^{(m)}}{\Delta d_n^{(m)}} \right\rceil - 1.$$

Alternatively, we may also specify the 1-D B-spline density, i.e., the number of 1-D B-spline atoms per unit length in the mth coordinate direction, $\kappa_{n,m} \triangleq (\tilde{K}_{n,m})/d_n^{(m)}$. Here, $\tilde{K}_{n,m}$ follows as

$$\tilde{K}_{n,m} = \left\lceil \kappa_{n,m}\, d_n^{(m)} \right\rceil. \tag{15}$$

For the Fourier dictionary, according to Section IV-A, there are $2\tilde{K}_{n,m}+1$ 1-D Fourier atoms in the mth coordinate direction. These atoms involve $\tilde{K}_{n,m}+1$ different frequencies (including frequency 0), which are given by $\nu_{\tilde{k}} \triangleq (\tilde{k}-1)/d_n^{(m)}$, $\tilde{k}=1,\ldots,\tilde{K}_m+1$. The bandwidth is given by the maximum frequency, $\nu_{\tilde{K}_{n,m}+1} = \tilde{K}_{n,m}/d_n^{(m)}$. If $\nu_{\tilde{K}_{n,m}+1}$ is specified, $\tilde{K}_{n,m}$ is obtained as

$$\tilde{K}_{n,m} = \left\lfloor \nu_{\tilde{K}_{n,m}+1}\, d_n^{(m)} \right\rfloor.$$

We also recall from Section IV that the overall dictionary size, i.e., the number of M-D atoms $\psi_k(\mathbf{x})$, $k=1,\ldots,K_n$, is given by $K_n = \prod_{m=1}^{M} \tilde{K}_{n,m}$ in the B-spline case and $K_n = \prod_{m=1}^{M}(2\tilde{K}_{n,m}+1)$ in the Fourier case. Fig. 3 illustrates the covering of the ROI $\mathcal{R}_n$ with B-spline atoms using two different densities of the B-spline atoms. A higher density enables a more accurate approximation of the log-LLF $\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ within $\mathcal{R}_n$ but also implies a larger dictionary size K_n and, potentially, a higher communication cost.

VI. Binary Coefficient Representation

According to Section II-C, the iterated LC coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$ that are communicated between the sensors are initialized at $i=0$ by the local expansion coefficients $\alpha_n^{(s,k)}$. Let L_s denote the number of $\alpha_n^{(s,k)}$ at sensor s. If the OMP method was used for calculating the $\alpha_n^{(s,k)}$, then, according to Section III-B, L_s is given by the number of OMP iterations. If the LS method was used, then, according to Section III-A, the number of $\alpha_n^{(s,k)}$ is nominally equal to the total number of dictionary atoms at time n, K_n. However, this number can be reduced by retaining only $L_s \leq K_n$ *dominant* expansion coefficients $\alpha_n^{(s,k)}$, i.e., those with largest absolute values or those whose absolute values are above a specified positive threshold. We note that L_s may depend on n, which is not shown by our notation.

In LC iteration i, each sensor s broadcasts its iterated coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$, $k = 1, \ldots, K_n$ to the neighboring sensors $s' \in \mathcal{N}_s$. In practice, this is done in a binary format. If each $\hat{\beta}_n^{(k,s)}[i]$ is represented by a bit sequence of length n_b, then the total number of bits broadcast by sensor s in LC iteration i is $N_\mathrm{naive} = K_n n_\mathrm{b}$. However, this communication cost can be significantly reduced by broadcasting only the *nonzero* $\hat{\beta}_n^{(k,s)}[i]$ plus additional bits that indicate their indices k and thus enable a correct interpretation of the stream of length-n_b bit sequences at the receiving sensors. Let $L_s[i]$ denote the number of nonzero $\hat{\beta}_n^{(k,s)}[i]$. Before the first consensus iteration $i = 1$, the coefficient estimates are initialized as $\hat{\beta}_n^{(k,s)}[0] = \alpha_n^{(s,k)}$, and thus $L_s[0] = L_s$. In the course of the consensus iterations $i = 1, 2, \ldots, I$, $L_s[i]$ will generally increase beyond L_s. This is because at different sensors s, the sets $\{k_l\}$ of indices k for which the $\hat{\beta}_n^{(k,s)}[i]$ are nonzero are typically not exactly equal, and thus the consensus update operation in (7) will cause some of the zero $\hat{\beta}_n^{(k,s)}[i]$ to become nonzero. As a consequence, the number of nonzero iterated coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$ broadcast by each sensor will be somewhat larger than L_s (but still much smaller than the total number of coefficients, K_n).

In the following, we discuss three different methods for encoding the indices k of the nonzero $\hat{\beta}_n^{(k,s)}[i]$ in a binary format. Our first method, termed the *indicator method*, uses an indicator bit vector of length K_n whose kth bit is 1 if $\hat{\beta}_n^{(k,s)}[i]$ is nonzero and 0 otherwise. This indicator bit vector is broadcast to the neighboring sensors in addition to the $L_s[i]$ n_b-bit sequences representing the nonzero $\hat{\beta}_n^{(k,s)}[i]$. Accordingly, the total number of bits broadcast by sensor s in LC iteration i is

$$N_{\mathrm{indicator},i}^{(s)} = L_s[i]\, n_\mathrm{b} + K_n\,. \tag{16}$$

This is smaller than $N_\mathrm{naive} = K_n n_\mathrm{b}$ if and only if $L_s[i] < K_n(1 - 1/n_\mathrm{b})$.

An alternative method, termed the *label method*, transmits along with each n_b-bit sequence representing a nonzero $\hat{\beta}_n^{(k,s)}[i]$ a binary "label" sequence encoding the index k. Since there are K_n possible indices k, each label sequence consists of $\lceil \log_2(K_n) \rceil$ bits. Therefore, the total number of bits broadcast by sensor s in LC iteration i is

$$N_{\mathrm{label},i}^{(s)} = L_s[i]\,\big(n_\mathrm{b} + \lceil \log_2(K_n) \rceil\big)\,. \tag{17}$$

This is smaller than $N_\mathrm{naive} = K_n n_\mathrm{b}$ if and only if $L_s[i] < K_n / \big(1 + \lceil \log_2(K_n) \rceil / n_\mathrm{b}\big)$, and smaller than $N_{\mathrm{indicator},i}^{(s)}$ if and only if $L_s[i] < K_n / \lceil \log_2(K_n) \rceil$.

A third coding method, termed the *hyperrectangle method*, presupposes localized atoms and is thus specifically suited to the B-spline dictionary. As described in Section IV-B, the B-spline atom $\psi_k(\mathbf{x})$ corresponding to $\hat{\beta}_n^{(k,s)}[i]$ is localized around some point $\mathbf{x}^{(k)}$ on a regular M-D grid within the ROI $\mathcal{R}_n$. This grid point can be alternatively indexed by the M-D index $\tilde{\mathbf{k}} \triangleq (\tilde{k}_1 \cdots \tilde{k}_M)^\mathrm{T}$ with $\tilde{k}_m \in \{1, \ldots, \tilde{K}_m\}$ (see (13)). Due to the localization of the atoms $\psi_k(\mathbf{x})$, the grid points $\mathbf{x}^{(k)}$ corresponding to the *nonzero* $\hat{\beta}_n^{(k,s)}[i]$ can be expected to lie in a relatively small subregion of $\mathcal{R}_n$. Equivalently, the associated M-D indices $\tilde{\mathbf{k}}$ are located in a relatively small subset of the total M-D index space. This subset is an M-D "discrete hyperrectangle" $\mathcal{K}_n^{(s)}[i]$ that consists of all $\tilde{\mathbf{k}}$ such that $\tilde{k}_m \in \{\tilde{l}_{n,m}^{(s)}[i], \ldots, \tilde{l}_{n,m}^{(s)}[i] + \Delta\tilde{k}_{n,m}^{(s)}[i]\}$ for $m = 1, \ldots, M$. Here, $\tilde{l}_{n,m}^{(s)}[i]$ and $\tilde{l}_{n,m}^{(s)}[i] + \Delta\tilde{k}_{n,m}^{(s)}[i]$ are, respectively, the minimum and maximum mth-coordinate index $\tilde{k}_m$ of any nonzero $\hat{\beta}_n^{(k,s)}[i]$. The number of different $\hat{\beta}_n^{(k,s)}[i]$ contained in $\mathcal{K}_n^{(s)}[i]$ is $|\mathcal{K}_n^{(s)}[i]| = \prod_{m=1}^{M}(\Delta\tilde{k}_{n,m}^{(s)}[i] + 1)$; out of these, $L_s[i]$ are nonzero. The basic geometry is visualized in Fig. 4.

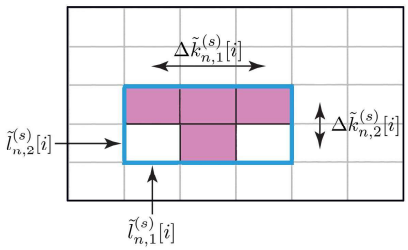


Fig. 4: Illustration (for dimension $M = 2$) of the ROI $\mathcal{R}_n$ and the hyperrectangle $\mathcal{K}_n^{(s)}[i]$ (blue rectangle) that contains the 2-D indices $\tilde{\mathbf{k}} = (\tilde{k}_1\ \tilde{k}_2)^\mathrm{T}$ of the nonzero coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$ (filled pink rectangles).

Sensor s then broadcasts the $L_s[i]$ nonzero $\hat{\beta}_n^{(k,s)}[i]$ using $L_s[i]$ binary sequences of length n_b, and—adopting, e.g., the strategy of the indicator method—a binary indicator vector of length $|\mathcal{K}_n^{(s)}[i]|$. This requires $L_s[i]\, n_\mathrm{b} + |\mathcal{K}_n^{(s)}[i]|$ bits. In addition, sensor s informs the neighboring sensors about the position and extent of $\mathcal{K}_n^{(s)}[i]$ within $\mathcal{R}_n$ by broadcasting binary representations of the "minimum vertex vector" $\tilde{\boldsymbol{l}}_n^{(s)}[i] \triangleq \big(\tilde{l}_{n,1}^{(s)}[i] \cdots \tilde{l}_{n,M}^{(s)}[i]\big)^\mathrm{T}$ and the "extent vector" $\boldsymbol{\Delta}_n^{(s)}[i] \triangleq \big(\Delta\tilde{k}_{n,1}^{(s)}[i] \cdots \Delta\tilde{k}_{n,M}^{(s)}[i]\big)^\mathrm{T}$. Because $\tilde{\boldsymbol{l}}_n^{(s)}[i]$ may be any one of the $K_n = \prod_{m=1}^{M}\tilde{K}_{n,m}$ possible M-D indices $\tilde{\mathbf{k}}$, $\lceil \log_2(K_n) \rceil = \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m}) \big\rceil$ bits are required to represent it. Furthermore, $\Delta\tilde{k}_{n,m}^{(s)}[i]$ must lie in the set $\{1, \ldots, \tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]\}$. Thus, $N_\Delta \triangleq \prod_{m=1}^{M}(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i])$ different extent vectors $\boldsymbol{\Delta}_n^{(s)}[i]$ are possible, which means that $\lceil \log_2(N_\Delta) \rceil = \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]) \big\rceil$ bits are required to represent $\boldsymbol{\Delta}_n^{(s)}[i]$. In summary, the total number of bits broadcast by sensor s in LC iteration i is given by $N_{\mathrm{hyperrect},i}^{(s)} = L_s[i]\, n_\mathrm{b} + |\mathcal{K}_n^{(s)}[i]| + \lceil \log_2(K_n) \rceil + \lceil \log_2(N_\Delta) \rceil$ or, in more detail,

$$N_{\mathrm{hyperrect},i}^{(s)} = L_s[i]\, n_\mathrm{b} + \prod_{m=1}^{M}(\Delta\tilde{k}_{n,m}^{(s)}[i] + 1) + \left\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m}) \right\rceil + \left\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]) \right\rceil.$$

Comparing with (16) and (17), we see that $N_{\mathrm{hyperrect},i}^{(s)}$ is smaller than $N_{\mathrm{indicator},i}^{(s)}$ and $N_{\mathrm{label},i}^{(s)}$ if and only if $\prod_{m=1}^{M}(\Delta\tilde{k}_{n,m}^{(s)}[i] + 1) + \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m}) \big\rceil + \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]) \big\rceil$ is smaller than K_n in the former case and $L_s[i] \lceil \log_2(K_n) \rceil$ in the latter case. This essentially amounts to the condition that the coordinate extents $\Delta\tilde{k}_{n,m}^{(s)}[i]$ are small enough.

VII. LC 2.0 for the Distributed PDAF

So far, we presented LC 2.0 for distributed particle filtering. In this section, extending our conference publication [8], we

consider the use of LC 2.0 within a distributed implementation of the probabilistic data association filter (PDAF).

A. Measurement Model, LLF, and GLF

The measurement model underlying the PDAF extends the measurement model of Section II-A to multiple measurements per sensor, missed detections, clutter, and a related measurement-origin uncertainty [2], [7]. At each time $n \geq 1$, each sensor $s = 1, \ldots, S$ now produces $W_n^{(s)}$ measurements $\mathbf{z}_{n,w}^{(s)}$, $w = 1, \ldots, W_n^{(s)}$, where $W_n^{(s)}$ may also be zero. These measurements include at most one target-generated measurement, the other measurements being clutter (false alarms). However, the sensor does not know the origins (target or clutter) of the measurements. We make the following assumptions: (i) At time n, sensor s "detects" the target, in the sense of producing a target-generated measurement, with probability $P_{\mathrm{d}}^{(s)}$. This measurement is distributed according to the conditional pdf $f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}\big|\mathbf{x}_n\big)$. (ii) The clutter measurements are independent and identically distributed (iid) with pdf $f_{\mathrm{c}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}\big)$. (iii) The number of clutter measurements at sensor s is Poisson distributed with mean $\mu^{(s)}$. Using these assumptions, it was shown in [2, Sec. 4.5] that the LLF at sensor s is given by

$$f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) = l_{n,0}^{(s)} + \sum_{w=1}^{W_n^{(s)}} l_{n,w}^{(s)}(\mathbf{x}_n), \tag{18}$$

with the nonnegative constant "floor" component

$$l_{n,0}^{(s)} \triangleq C\big(\mathbf{z}_n^{(s)}\big)\big(1-P_{\mathrm{d}}^{(s)}\big)\mu^{(s)} \tag{19}$$

and the $W_n^{(s)}$ nonnegative measurement-related components

$$l_{n,w}^{(s)}(\mathbf{x}_n) \triangleq C\big(\mathbf{z}_n^{(s)}\big) P_{\mathrm{d}}^{(s)} \frac{f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}\big|\mathbf{x}_n\big)}{f_{\mathrm{c}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}\big)}, \quad w = 1, \ldots, W_n^{(s)}. \tag{20}$$

Here, $\mathbf{z}_n^{(s)} \triangleq \big(\mathbf{z}_{n,1}^{(s)\mathrm{T}} \cdots \mathbf{z}_{n,W_n^{(s)}}^{(s)\mathrm{T}}\big)^{\mathrm{T}}$ comprises all the measurements at sensor s, and $C\big(\mathbf{z}_n^{(s)}\big)$ is a normalization constant. In the absence of missed detections and clutter (i.e., $P_{\mathrm{d}}^{(s)} = 1$, $W_n^{(s)} = 1$, $\mu^{(s)} = 0$), the LLF in (18)–(20) would simplify to $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) = f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,1}^{(s)}\big|\mathbf{x}_n\big)$, i.e., the floor component would be zero and there would be only one measurement-related component, which is target-generated; this equals the measurement model of Section II-A.

We consider a generalization of the original PDAF to nonlinear and non-Gaussian state-space models. This generalized PDAF is a particle filter based on the likelihood function in (18)–(20) [2], [7]. For a distributed implementation, each sensor s runs a local particle filter as described in Section II-B. We again assume that the measurements $\mathbf{z}_n^{(s)}$ at different sensors are conditionally independent given $\mathbf{x}_n$. Then, the GLF is the product of the LLFs (see (1)), and both the LC and the proposed LC 2.0 can again be used for a distributed calculation of the GLF approximations $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ involved in the update step (3). However, a difference from conventional distributed particle filtering is that now we also have to take into account the constant floor component $l_{n,0}^{(s)}$ of the LLF (18).

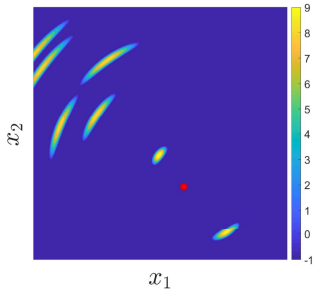


Fig. 5: Example (simulation result, using one sensor at the position indicated by the red bullet) of a log-LLF for dimension $M = 2$. This log-LLF comprises seven measurement-related components, whereof one is target-generated and the remaining six are clutter. The dark blue background corresponds to the floor component $\lambda_{n,0}^{(s)}$, which equals the minimum of the log-LLF.

B. Splitting Off the Floor

We now propose a modification of the LC that promotes a sparse representation of the LLF (18) by splitting off the floor component $l_{n,0}^{(s)}$. For practically relevant sensors, the measurement-related components $l_{n,w}^{(s)}(\mathbf{x}_n)$ are effectively supported in subregions of $\mathbb{R}^M$. This implies that in the complementary subregion, according to (18), the LLF $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ is effectively equal to the floor component $l_{n,0}^{(s)}$. Since moreover the measurement-related LLF part $\sum_{w=1}^{W_n^{(s)}} l_{n,w}^{(s)}(\mathbf{x}_n)$ is nonnegative, it follows that the floor component $l_{n,0}^{(s)}$ is approximately equal to the minimum of the LLF $f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$, i.e., we effectively have $l_{n,0}^{(s)} = \min_{\mathbf{x}_n \in \mathbb{R}^M} f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$, and consequently

$$f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) \geq l_{n,0}^{(s)}. \tag{21}$$

Let us now consider the log-LLF $\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$ and, in particular, its minimum

$$\lambda_{n,0}^{(s)} \triangleq \min_{\mathbf{x}_n \in \mathbb{R}^M} \log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big). \tag{22}$$

Using the fact that log is a strictly increasing function, we have

$$\lambda_{n,0}^{(s)} = \log\Big(\min_{\mathbf{x}_n \in \mathbb{R}^M} f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)\Big) = \log l_{n,0}^{(s)}, \tag{23}$$

and we conclude from (21) that $\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) \geq \log l_{n,0}^{(s)}$ or, equivalently,

$$\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) \geq \lambda_{n,0}^{(s)}. \tag{24}$$

It follows from (24) that the log-LLF can be written as

$$\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) = \lambda_{n,0}^{(s)} + \lambda_n^{(s)}(\mathbf{x}_n), \tag{25}$$

with a nonnegative function $\lambda_n^{(s)}(\mathbf{x}_n)$. (Note that, by contrast, the "log-LLF floor" $\lambda_{n,0}^{(s)}$ may be negative.) An example of a log-LLF is shown in Fig. 5. Inserting (25) into (4) yields

$$L_n(\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S}\big(\lambda_{n,0}^{(s)} + \lambda_n^{(s)}(\mathbf{x}_n)\big) = \bar{\lambda}_{n,0} + \bar{\lambda}_n(\mathbf{x}_n), \tag{26}$$

with the *log-GLF floor* $\bar{\lambda}_{n,0}$ and the *floorless log-GLF* $\bar{\lambda}_n(\mathbf{x}_n)$

defined as

$$\bar{\lambda}_{n,0} \triangleq \frac{1}{S}\sum_{s=1}^{S}\lambda_{n,0}^{(s)}, \quad \bar{\lambda}_n(\mathbf{x}_n) \triangleq \frac{1}{S}\sum_{s=1}^{S}\lambda_n^{(s)}(\mathbf{x}_n). \tag{27}$$

According to Section II-C, the LC-based distributed calculation of the approximate GLF $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ then amounts to a distributed calculation of the function $L_n(\mathbf{x}_n) = \bar{\lambda}_{n,0} + \bar{\lambda}_n(\mathbf{x}_n)$. Obtaining a sparse LC expansion is facilitated by splitting off the log-GLF floor $\bar{\lambda}_{n,0}$ from $L_n(\mathbf{x}_n)$ and performing separate distributed calculations of $\bar{\lambda}_{n,0}$ and $\bar{\lambda}_n(\mathbf{x}_n)$. Because $\bar{\lambda}_{n,0}$ is (by (27)) the average of the S *scalars* $\lambda_{n,0}^{(s)}$, it can be approximated in a distributed manner via a single instance of the average consensus algorithm. The floorless log-GLF $\bar{\lambda}_n(\mathbf{x}_n)$, on the other hand, is the average of the S *functions* $\lambda_n^{(s)}(\mathbf{x}_n)$; it can be approximated in a distributed manner via the LC. Thus, the LC is used to calculate only the floorless log-LLF $\bar{\lambda}_n(\mathbf{x}_n)$ rather than the entire function $L_n(\mathbf{x}_n)$.

For a practical implementation, each sensor s first determines its log-LLF floor $\lambda_{n,0}^{(s)}$ by finding the minimum of $\log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)$. (In principle, the minimum can be obtained from the closed-form expression (19); however, we observed that a numerical computation according to (22) yielded a slightly better tracking performance.) Then, a single instance of the average consensus algorithm is used to calculate the log-GLF floor $\bar{\lambda}_{n,0} = \frac{1}{S}\sum_{s=1}^{S}\lambda_{n,0}^{(s)}$. Next, each sensor s calculates its floorless log-LLF according to $\lambda_n^{(s)}(\mathbf{x}_n) = \log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) - \lambda_{n,0}^{(s)}$ (see (25)). The LC is now used to calculate the floorless log-GLF $\bar{\lambda}_n(\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S}\lambda_n^{(s)}(\mathbf{x}_n)$. Finally, each sensor calculates $L_n(\mathbf{x}_n)$ according to (26), i.e., by adding its local estimates of $\bar{\lambda}_{n,0}$ and $\bar{\lambda}_n(\mathbf{x}_n)$.

C. Using LC 2.0

A simplification of the strategy described above is possible when using LC 2.0—more specifically, when using a B-spline dictionary and the adaptively determined ROI. Here, most of the support of the log-LLF floor is removed implicitly by approximating the log-LLF only on the ROI $\mathcal{R}_n$. Moreover, since according to Section V-A $\mathcal{R}_n$ is a slightly extended version of the effective support of the global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$, it typically contains the target-generated component of the log-LLF but excludes most of the clutter components. When a B-spline dictionary is used on $\mathcal{R}_n$, then, due to the localization of the B-spline atoms, one obtains a sparse LC expansion without splitting off the log-LLF floor.

On the other hand, also $\mathcal{R}_n$ typically includes subregions that do not contain measurement-related LLF components and thus belong to the support of the LLF floor. Therefore, an even better sparsity can generally be obtained by splitting off the log-LLF floor *within* $\mathcal{R}_n$ and approximating the floorless log-GLF $\bar{\lambda}_n(\mathbf{x}_n)$ on $\mathcal{R}_n$. Differently from (23), the log-LLF floor is now determined only on $\mathcal{R}_n$, i.e., $\tilde{\lambda}_{n,0}^{(s)} \triangleq \log\big(\min_{\mathbf{x}_n\in\mathcal{R}_n} f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big)\big)$, and the calculation of the floorless log-LLF $\tilde{\lambda}_n^{(s)}(\mathbf{x}_n) \triangleq \log f\big(\mathbf{z}_n^{(s)}\big|\mathbf{x}_n\big) - \tilde{\lambda}_{n,0}^{(s)}$ and the B-spline expansion of $\tilde{\lambda}_n^{(s)}(\mathbf{x}_n)$ are performed on $\mathcal{R}_n$.

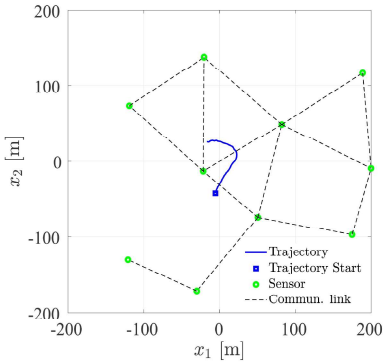


Fig. 6: Surveillance region, sensor network, and target trajectory used in our simulation.

VIII. Simulation Results

In this section, we evaluate and compare the performance and communication cost of LC 2.0 relative to the conventional LC. For a detailed understanding of LC 2.0, we study the effects of the different aspects of LC 2.0—OMP, B-spline dictionary, binary representation, uniform log-LLF sampling, and adaptive ROI—separately.

A. Simulation Setup

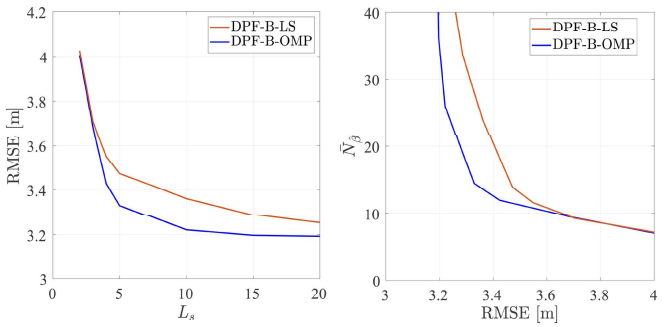
We simulated a target moving in the 2-D plane. The target state is $\mathbf{x}_n = (x_{n,1}\ x_{n,2}\ \dot{x}_{n,1}\ \dot{x}_{n,2})^{\mathrm{T}}$, where $x_{n,1}$, $x_{n,2}$ and $\dot{x}_{n,1}$, $\dot{x}_{n,2}$ are the components of the target's position and velocity, respectively. The surveillance area is $[-200\,\mathrm{m}, 200\,\mathrm{m}] \times [-200\,\mathrm{m}, 200\,\mathrm{m}]$. The evolution of the state $\mathbf{x}_n$ is modeled as $\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{\Gamma}\mathbf{u}_n$, where the matrices $\mathbf{F} \in \mathbb{R}^{4\times4}$ and $\mathbf{\Gamma} \in \mathbb{R}^{4\times2}$ are defined in [33], involving a time step parameter T that is chosen as 1s, and the driving noise $\mathbf{u}_n \in \mathbb{R}^2$ is iid, zero-mean, and Gaussian with covariance matrix $\sigma_u^2\mathbf{I}_2$, where $\sigma_u = 1/3$ m/s^2. It follows that the state-transition pdf $f(\mathbf{x}_n|\mathbf{x}_{n-1})$ is Gaussian with mean $\mathbf{F}\mathbf{x}_{n-1}$ and covariance matrix $\sigma_u^2\mathbf{\Gamma}\mathbf{\Gamma}^{\mathrm{T}}$. The sensor network consists of $S = 10$ sensors. Fig. 6 shows the surveillance area, the sensor network, and the target trajectory used in our simulations.

Each sensor s produces a range (distance) and bearing (angle) measurement given by

$$\mathbf{z}_n^{(s)} = \big(\|\tilde{\mathbf{x}}_n - \mathbf{p}^{(s)}\| \ \ \varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)})\big)^{\mathrm{T}} + \mathbf{v}_n^{(s)}, \tag{28}$$

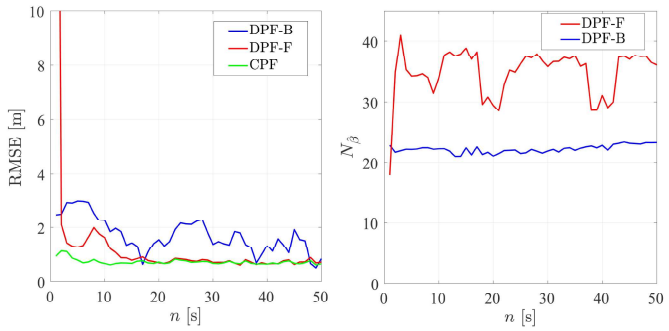
where $\tilde{\mathbf{x}}_n \triangleq (x_{n,1}\ x_{n,2})^{\mathrm{T}}$ is the position of the target, $\mathbf{p}^{(s)}$ is the position of sensor s, $\varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)})$ is the angle of $\tilde{\mathbf{x}}_n$ relative to $\mathbf{p}^{(s)}$, and $\mathbf{v}_n^{(s)}$ is iid, zero-mean, Gaussian measurement noise with covariance matrix $\mathbf{C}_v = \mathrm{diag}\{\sigma_{\mathrm{r}}^2, \sigma_{\mathrm{b}}^2\}$, where $\sigma_{\mathrm{r}} = 5/3$ m and $\sigma_{\mathrm{b}} = 10/3°$. Because $\mathbf{z}_n^{(s)}$ depends only on the position $\tilde{\mathbf{x}}_n$, the LLF is effectively given by $f\big(\mathbf{z}_n^{(s)}\big|\tilde{\mathbf{x}}_n\big)$, which implies that our effective state-space dimension is $M = 2$. A modified measurement model will be used in Section VIII-H.

Each local particle filter uses $J = 10000$ particles. For initialization of the particles at time $n = 0$, the position $\tilde{\mathbf{x}}_n$ is sampled uniformly at random in the surveillance area, the target speed is sampled from a truncated Gaussian distribution with mean 2 m/s^2 and standard deviation $1/3$ m/s^2, and the target heading is sampled from a uniform distribution on $(-180°, 180°]$. The LC employs $I = 20$ consensus iterations.



(a) (b)

Fig. 7: Comparison of DPFs with OMP-based and LS-based calculation of the local expansion coefficients, using a B-spline dictionary and uniform sampling in the entire surveillance area (these DPFs are abbreviated as DPF-B-OMP and DPF-B-LS, respectively): (a) Time-averaged localization RMSE versus number of nonzero local expansion coefficients, L_s, (b) average number of nonzero coefficient estimates, $\bar{N}_{\hat{\beta}}$, versus time-averaged localization RMSE.



(a) (b)

Fig. 8: Comparison of DPFs using a B-spline dictionary and a Fourier dictionary, both covering the entire surveillance area: (a) Localization RMSE versus time, (b) average number of nonzero coefficient estimates, $N_{\hat{\beta}}$, versus time.

The tracking accuracy of the DPF is measured by the localization root mean square error (RMSE) and the track loss percentage, based on 100 simulation runs performed over 50 time steps n (corresponding to a duration of 50 s). The localization RMSE is averaged over all sensors and all successful simulation runs. Here, a simulation run is considered successful if, for all time steps $n \geq 11$, the localization RMSE is smaller than 5 m; otherwise it is considered a track loss and included in the calculation of the track loss percentage.

B. OMP

First, we demonstrate the savings in communication that are achieved by the OMP-based calculation of the local expansion coefficients described in Section III-B relative to the LS-based calculation employed by the conventional LC. We only consider a B-spline dictionary because the benefits of using the OMP in conjunction with a Fourier dictionary were already discussed in [8]. Our B-spline dictionary comprises $K = 400$ atoms located uniformly in the entire surveillance area. For calculating the local expansion coefficients, we used uniform sampling of the log-LLF on the entire surveillance area (see Section III-C). That is, at this point, we do not use the adaptive ROI described in Section V. Fig. 7a shows the time-averaged localization RMSE (averaged over all time steps $n = 1, \ldots, 50$) versus the number of nonzero local expansion coefficients, $L_s \in \{2, \ldots, 20\}$. Here, L_s was chosen identically for each sensor s; in the OMP case, it equals the number of OMP iterations while in the LS case, we retained the L_s local coefficients with the largest absolute values. We conclude from Fig. 7a that for $L_s \geq 4$, OMP leads to a smaller localization RMSE than LS. Conversely, using OMP, a given RMSE is obtained with a smaller L_s than using LS. For example, an RMSE of about 3.3 m is obtained with $L_s = 15$ when using LS as opposed to only $L_s = 6$ when using OMP. To complete the picture, we note that neither OMP nor LS produced any track losses.

The smaller values of L_s required by the OMP translate into savings in the number of nonzero coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$. Fig. 7b shows the average number of nonzero coefficient estimates, $\bar{N}_{\hat{\beta}}$ (averaged over all LC iterations i, all sensors s, and all time steps n) versus the RMSE. For example, the average number of nonzero coefficient estimates that a sensor needs to broadcast for an RMSE of 3.3 m is reduced by about one half if the OMP method is used instead of the LS method. Because of this superiority of the OMP, we will not further consider the LS method in our subsequent simulations unless stated otherwise.

C. B-spline Dictionary

Next, we compare DPFs using a B-spline dictionary and a Fourier dictionary (abbreviated DPF-B and DPF-F, respectively). DPF-B uses $\tilde{K}_m = 40$ B-spline atoms in each coordinate direction $m = 1, 2$ and hence $K = 1600$ B-spline atoms in total, which are regularly spaced in the entire surveillance area. DPF-F uses a Fourier dictionary with $\tilde{K}_m = 20$ frequencies in each coordinate direction and hence $K = 1681$ Fourier atoms in total. Note that K is similar in DPF-B and DPF-F. In DPF-B, the log-LLF is sampled uniformly in the entire surveillance area using $Q_n = 160000$ samples, whereas in DPF-F, it is sampled at the $J = 10000$ particles; these sampling schemes produce the best results in the respective cases. (We note that Q_n can be chosen significantly smaller when the adaptive ROI is implemented, see Section VIII-F.) Both DPF-B and DPF-F use the OMP with $L_s = 5$ iterations.

Fig. 8a shows the localization RMSE versus time obtained with DPF-B, with DPF-F, and—as a performance benchmark—with a centralized multisensor particle filter (abbreviated CPF). For $n = 1$, the RMSE of DPF-F is 34 m and thus much higher than that of DPF-B; this can be explained by the fact that initially the particles are spread out over the entire surveillance area and, thus, not sufficiently concentrated. For $n \geq 2$, on the other hand, the RMSE of DPF-F is typically lower than that of DPF-B and effectively equals that of CPF for $n \geq 13$. However, the track loss results are contrary to the RMSE results: the track loss percentage of DPF-F was measured as 2.2, whereas DPF-B did not produce any track losses. Fig. 8b shows the number of nonzero coefficient estimates, $N_{\hat{\beta}}$, broadcast on average by a sensor during one LC iteration. The time-average of $N_{\hat{\beta}}$ is about 35 for DPF-F but

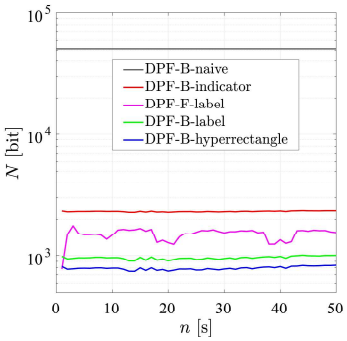


Fig. 9: Average binary communication costs of DPF-B variants using different binary coding methods, as well as of DPF-F using the label method.

only 22 for DPF-B, corresponding to a reduction by about 33%. This reduction due to the use of the B-spline dictionary will be seen to be even larger when the adaptive ROI is implemented, see Section VIII-F. Furthermore, $N_{\hat{\beta}}$ is seen to be fairly constant in the case of DPF-B (which implies an approximately constant communication cost) whereas it exhibits large time variations in the case of DPF-F.

D. Binary Representation

Our next simulation compares DPF-B variants using the binary coding methods presented in Section VI. All these DPF-B variants employ a binary wordlength of $n_{\mathrm{b}} = 32$ (the standard floating point format) and $K = 1600$ B-spline atoms in the entire surveillance area. Fig. 9 shows the binary communication costs N_{naive} as well as $N^{(s)}_{\mathrm{indicator},i}$, $N^{(s)}_{\mathrm{label},i}$, and $N^{(s)}_{\mathrm{hyperrect},i}$ averaged over all sensors s and all LC iterations i. It can be seen that the binary communication cost is smallest for the hyperrectangle method (its time-average is only 790.71 bit), somewhat higher for the label method, and much higher for the indicator method. However, all are significantly lower than the communication cost of naive coding, $N_{\mathrm{naive}} = Kn_{\mathrm{b}} = 51200$ bit. We conclude that the proposed binary coding methods lead to large savings in communication. The inferiority of the indicator method relative to the label and hyperrectangle methods is due to the large length of the binary indicator vector (which comprises $K = 1600$ bits) and will be seen in Section VIII-F to be less pronounced when the adaptive ROI is implemented.

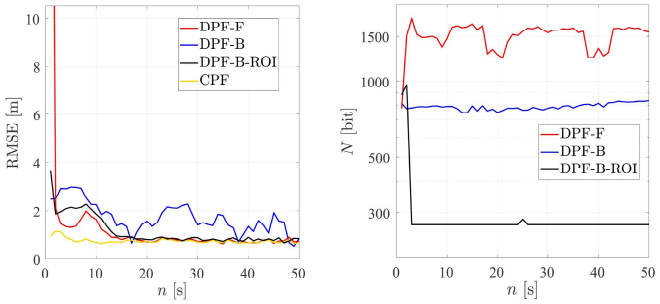
Fig. 9 also considers DPF-F with binary coding using the label method, which is the most efficient coding method for DPF-F. It is seen that the binary communication cost of this DPF-F variant is about twice that of DPF-B using the hyperrectangle method. Thus, using the B-spline dictionary instead of the Fourier dictionary reduces the binary communication cost by approximately one half.

E. Uniform Log-LLF Sampling

As mentioned in Section III-C, successful use of the B-spline dictionary requires that the log-LLF is sampled uniformly rather than at the particles. We now verify this fact and study the impact of the number of samples Q on the tracking performance. We consider two DPF-B variants, abbreviated as DPF-B-P(J) and DPF-B-U(Q), in which the log-LLF is sampled either at the particles using $J = 10000$ or 100000 particles or uniformly in the entire surveillance area using $Q = 1600$, 10000, 40000, or 160000 samples. Both DPF-B variants use $K = 1600$ B-spline atoms in the entire surveillance area. Table I presents the time-averaged localization RMSE and the track loss percentage ρ. It is seen that sampling of the log-LLF at the particles results in $\rho = 100\%$ (thus, there are no "successful" simulation runs from which to calculate the RMSE). By contrast, uniform sampling yields excellent performance, with $\rho = 0.2\%$ or 0% and RMSE around 1.7 m for $Q = 10000$ or larger.

TABLE I: Tracking performance of DPF-B variants using log-LLF sampling at the particles or uniformly in the entire surveillance area.

DPF	RMSE [m]	ρ [%]
DPF-B-P(10000)	N/A	100
DPF-B-P(100000)	N/A	100
DPF-B-U(1600)	2.35	0.2
DPF-B-U(10000)	1.70	0.2
DPF-B-U(40000)	1.73	0.2
DPF-B-U(160000)	1.70	0



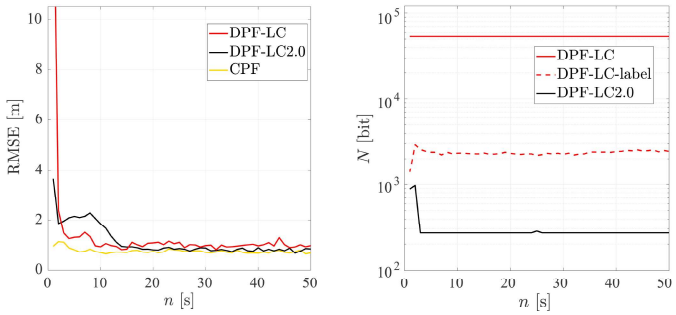
(a) (b)

Fig. 10: Comparison of DPF-B with and without adaptive ROI: (a) Localization RMSE, (b) average binary communication cost.

F. Adaptive ROI

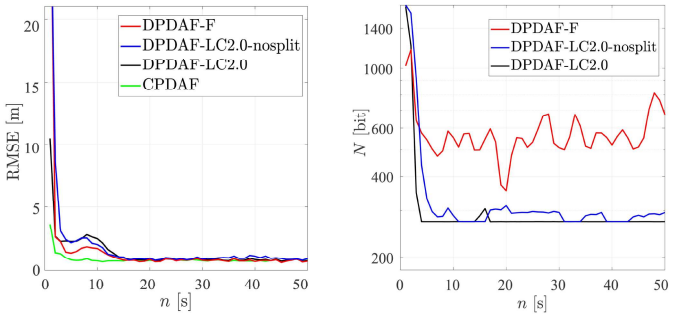
Next, we demonstrate the benefits of using the adaptive ROI introduced in Section V-A. We consider a DPF-B variant, designated DPF-B-ROI, in which the B-spline dictionary covers only the ROI $\mathcal{R}_n$. The parameter γ used in the calculation of $\mathcal{R}_n$ (see (14)) is 10. The numbers of B-spline atoms in the two coordinate directions, $\tilde{K}_{n,1}$ and $\tilde{K}_{n,2}$, are chosen adaptively according to (15) with $\kappa_{n,1} = \kappa_{n,2} = 1/5$, i.e., there are 5 atom centers per meter. The log-LLF is sampled uniformly on $\mathcal{R}_n$ with a density of one sample per meter; thus, the total number of samples is $Q_n = d^{(1)}_n d^{(2)}_n$, where the ROI interval lengths $d^{(1)}_n$ and $d^{(2)}_n$ are determined adaptively according to (14). For initialization of the local particle filters at $n = 1$, $\mathcal{R}_1$ is chosen as the entire surveillance area; furthermore, $\tilde{K}_{1,1} = \tilde{K}_{1,2} = 20$, corresponding to $K_1 = 400$ B-spline atoms, and $Q_1 = 160000$. For $n \geq 2$, the (adaptively determined) dictionary size K_n and number of log-LLF samples Q_n are much smaller than K_1 and Q_1, respectively: we observed $K_n = 4$ and Q_n around 1000 for almost all times and simulation runs.

Fig. 10 compares DPF-B-ROI and DPF-B, where the latter employs $K = 1600$ B-spline atoms in the entire surveillance area and uniform log-LLF sampling in the entire surveillance area using $Q_n = 160000$ samples. In addition, Fig. 10 shows



(a) (b)

Fig. 11: Comparison of a DPF using LC2.0 (identical to DPF-B-ROI in Fig. 10) with a DPF using the conventional LC and a DPF using the conventional LC enhanced by the label method for binary coding: (a) Localization RMSE, (b) average binary communication cost.



(a) (b)

Fig. 12: Comparison of a DPDAF using LC2.0 with and without floor splitting, a DPDAF using a Fourier dictionary on the entire surveillance area and the OMP [8], and a centralized PDAF: (a) Localization RMSE, (b) average binary communication cost.

the results of DPF-F (using $K = 1681$ Fourier atoms on the entire surveillance area and particle-based log-LLF sampling) and of CPF. All DPFs use the OMP with $L_s = \min\{5, K_n\}$ iterations (note that L_s cannot be larger than K_n). For binary coding, DPF-B-ROI and DPF-B use the hyperrectangle method and DPF-F uses the label method, all with binary wordlength $n_\mathrm{b} = 32$. One can see in Fig. 10a that the localization RMSE of DPF-B-ROI is typically significantly lower than that of DPF-B. Moreover, for n larger than about 7, the RMSE comes very close to that of DPF-F and, somewhat later, also to that of CPF. The measured track loss percentage was similar for DPF-B-ROI and DPF-B (0.2% and 0%, respectively) but higher for DPF-F (2.2%). Fig. 10b shows that for $n \geq 3$, the communication cost of DPF-B-ROI is only about 40% of that of DPF-B. We conclude that using the adaptive ROI results in both a significant gain in tracking accuracy and large savings in communication. The latter come in addition to the savings relative to DPF-F previously reported in Section VIII-C, which are again visible in Fig. 10b.

G. The Benefit of LC 2.0

After studying the individual aspects of LC 2.0 separately, we now demonstrate the total reduction of communication cost achieved with LC 2.0 relative to the conventional LC. Fig. 11 compares DPF-B-ROI (previously considered in Fig. 10, but now dubbed DPF-LC2.0) with a DPF using the conventional LC (dubbed DPF-LC). DPF-LC2.0 combines all the methodological constituents of LC 2.0—OMP, B-spline dictionary, efficient binary coding, uniform log-LLF sampling, and adaptive ROI—as previously described for DPF-B-ROI in Section VIII-F. By contrast, DPF-LC uses LS-based calculation of the local expansion coefficients, retaining the $L_s = 10$ dominant coefficients; a Fourier dictionary with $K = 1681$ Fourier atoms on the entire surveillance area; log-LLF sampling at the particles; and naive binary coding, i.e., each of the $K = 1681$ coefficients is represented by $n_\mathrm{b} = 32$ bits. Here, the parameters $L_s = 10$ and $K = 1681$ were chosen because they achieve a similar localization RMSE as DPF-LC2.0.

One can verify in Fig. 11a that, as intended, the localization RMSE of DPF-LC2.0 is similar to that of DPF-LC. On the other hand, the track loss percentage of DPF-LC2.0 and DPF-LC was measured as 0.2% and 4.2%, respectively, and thus the overall tracking performance of DPF-LC2.0 is considerably better than that of DPF-LC. Fig. 11b shows that the communication cost of DPF-LC2.0 is only about 0.5% of that of DPF-LC. A large part of this reduction is caused by the efficient binary coding. To demonstrate this fact, Fig. 11b also considers a DPF-LC variant, designated DPF-LC-label, that uses the label coding method instead of naive coding. The time-averaged communication cost was measured as $N = 53792$ bit for DPF-LC, $N = 2341.59$ bit for DPF-LC-label, and $N = 295.70$ bit for DPF-LC2.0. Thus, relative to DPF-LC, efficient binary coding reduces communication by a factor of about 20, and a further reduction by a factor of about 8 is achieved by the remaining constituents of LC 2.0 (OMP, B-spline dictionary, and adaptive ROI).

H. LC2.0 for the DPDAF

Finally, we consider the use of LC2.0 within a distributed PDAF (DPDAF) as proposed in Section VII. The simulation setup is as before, except that now there are multiple measurements per sensor with the following parameters (cf. Section VII-A): target detection probability $P_\mathrm{d}^{(s)} = 0.95$, clutter mean $\mu^{(s)} = 5$, clutter pdf $f_\mathrm{c}^{(s)}(\mathbf{z}_{n,w}^{(s)})$ uniform on the surveillance region. Fig. 12 compares the DPDAF using LC 2.0 (dubbed DPDAF-LC2.0) with the DPDAF using a Fourier dictionary on the entire surveillance area and the OMP as considered in [8] (DPDAF-F) and with a centralized PDAF (CPDAF). DPDAF-LC2.0 uses the adaptive ROI (with $\gamma = 10$) and a B-spline dictionary (with the number of atoms adaptively determined using spacing $\kappa_{n,1} = \kappa_{n,2} = 1/5$, and initialized as $K_1 = 400$), and it splits off the log-GLF floor within the ROI as described in Section VII-C. Both DPDAF-LC2.0 and DPDAF-F use $L_s = \min\{5, K_n\}$ OMP iterations and the best binary encoding method (hyperrectangle and label method, respectively). DPDAF-F uses $K = 441$ Fourier atoms, which was observed to be the minimal number of atoms yielding a track loss percentage ρ below 5%. Fig. 12 also considers a variant of DPDAF-LC2.0 that does not split off the log-GLF floor (dubbed DPDAF-LC2.0-nosplit).

We can see in Fig. 12a that for $n \geq 15$, the localization RMSE of all four methods is almost equal. At $n = 1$, the RMSE of DPDAF-LC2.0 is significantly lower than that of DPDAF-

F, whereas for n between 3 and 11, it is somewhat higher. The track loss percentage was measured as $\rho = 0.4\%$ for both DPDAF-LC2.0 and DPDAF-F. The RMSE of DPDAF-LC2.0-nosplit is higher than that of DPDAF-LC2.0 for $n = 1, 2$ and almost as high as that of DPDAF-F for $n = 1$. Moreover, the track loss percentage of DPDAF-LC2.0-nosplit was measured as 4.8% and is thus significantly higher than that of DPDAF-LC2.0. Hence, splitting off the log-GLF floor improves the tracking accuracy. Fig. 12b shows that the communication cost of DPDAF-LC2.0 is only about half that of DPDAF-F. This reduction is due to the use of the B-spline dictionary and the adaptive ROI. Furthermore, the communication cost of DPDAF-LC2.0 is seen to be lower than that of DPDAF-LC2.0-nosplit, especially for $n \leq 4$. Indeed, during this initial phase, splitting off the log-GLF floor yields a faster convergence of the ROI (we recall that at time $n = 1$, the ROI is initialized as the entire surveillance area) and, in turn, a lower communication cost.

IX. Conclusion

The likelihood consensus (LC) scheme enables approximately Bayes-optimal distributed target tracking in nonlinear and non-Gaussian sensor networks. We proposed an evolved "LC 2.0" scheme with significantly reduced communication cost. LC 2.0 incorporates several modifications of the original LC scheme including the use of the OMP and a B-spline dictionary, efficient binary representations, and a distributed adaptation of the region of interest. Simulation results demonstrated savings in communication of about a factor of 200, without a loss in tracking performance.

An interesting direction for future work is the application of the proposed LC 2.0 scheme to other distributed Bayesian filtering frameworks in which the global likelihood function factors into the local likelihood functions. In particular, the LC-based distributed Bernoulli filter presented in [23] can be easily adapted to LC 2.0.

Acknowledgments

The authors would like to thank Ondřej Klimeš and Rene Repp for early contributions to the development and numerical evaluation of LC 2.0.

References

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

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

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

[4] 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.

[5] 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.

[6] 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.

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

[8] R. Repp, P. Rajmic, F. Meyer, and F. Hlawatsch, "Target tracking using a distributed particle-PDA filter with sparsity-promoting likelihood consensus," in *Proc. IEEE SSP*, Freiburg im Breisgau, Germany, June 2018, pp. 653–657.

[9] M. J. Coates, "Distributed particle filters for sensor networks," in *Proc. IEEE/ACM IPSN-04*, Berkeley, CA, USA, Apr. 2004, pp. 99–107.

[10] V. Savic, H. Wymeersch, and S. Zazo, "Belief consensus algorithms for fast distributed target tracking in wireless sensor networks," *Signal Process.*, vol. 95, pp. 149–160, 2014.

[11] W. Song, Z. Wang, J. Wang, F. E. Alsaadi, and J. Shan, "Distributed auxiliary particle filtering with diffusion strategy for target tracking: A dynamic event-triggered approach," *IEEE Trans. Signal Process.*, vol. 69, pp. 328–340, 2021.

[12] A. Mohammadi and A. Asif, "Diffusive particle filtering for distributed multisensor estimation," in *Proc. IEEE ICASSP*, Shanghai, China, 2016, pp. 3801–3805.

[13] W. Xia, M. Sun, and Z. Zhou, "Diffusion collaborative feedback particle filter," *IEEE Signal Process. Lett.*, vol. 27, pp. 1185–1189, 2020.

[14] T. Ghirmai, "Distributed particle filter using Gaussian approximated likelihood function," in *Proc. CISS*, Princeton, NJ, USA, Mar. 2014, pp. 1–5.

[15] X. Zhong and A. B. Premkumar, "Particle filtering approaches for multiple acoustic source detection and 2-D direction of arrival estimation using a single acoustic vector sensor," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4719–4733, 2012.

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

[17] O. Hlinka, F. Hlawatsch, and P. M. Djurić, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 61–81, 2013.

[18] M. G. Bruno and S. S. Dias, "A Bayesian interpretation of distributed diffusion filtering algorithms [lecture notes]," *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 118–123, 2018.

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

[20] 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.

[21] H. Prautzsch, W. Boehm, and W. Paluszny, *Bézier and B-Spline Techniques*. Berlin, Germany: Springer, 2002.

[22] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, 1999.

[23] G. Papa, R. Repp, F. Meyer, P. Braca, and F. Hlawatsch, "Distributed Bernoulli filtering using likelihood consensus," *IEEE Trans. Signal Inform. Process. Netw.*, vol. 5, no. 2, pp. 218–233, 2019.

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

[25] 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.

[26] 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.

[27] 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.

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

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

[30] 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.

[31] Å. Björck, "Least squares methods," in *Handbook of Numerical Analysis*, P. G. Ciarlet and J. L. Lions, Eds. Amsterdam, The Netherlands: Elsevier, 1990, vol. 1, pp. 465–652.

[32] V. Yadav and M. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Proc. 45th Annu. Allerton Conf. Commun. Contr. Comput.*, Monticello, IL, USA, Sep. 2007, pp. 715–720.

[33] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: Wiley, 2001.