Signal Processing 215 (2024) 109259

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Distributed Bayesian target tracking with reduced communication: Likelihood consensus 2.0

E. Šauša [a], P. Rajmic [b,*], F. Hlawatsch [a]

[a] Institute of Telecommunications, TU Wien, Vienna, Austria
[b] Department of Telecommunications, Brno University of Technology, Brno, Czech Republic

ARTICLE INFO

Keywords:
Target tracking
Distributed particle filter
Likelihood consensus
Splines
Orthogonal matching pursuit
OMP
Sparsity
Distributed PDA filter

ABSTRACT

The likelihood consensus (LC) enables Bayesian target tracking in a decentralized sensor network with possibly nonlinear and non-Gaussian sensor characteristics. 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 a reduction of intersensor communication by a factor of about 190 can be obtained without compromising the tracking performance.

1. Introduction

1.1. Background

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 [4,5,9–18]. Besides DPF methods using the LC or other consensus-based information dissemination strategies [4,5,10,16], there are also DPF methods based on the diffusion strategy [11–13,17]. Diffusion-based 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 exchange also 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 [17].

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.

1.2. Contributions and paper organization

Here, we propose an evolved LC methodology – dubbed LC 2.0 – with significantly reduced communication cost. We introduce the following modifications and extensions of the LC scheme:

- A sparsity-promoting calculation of the LC expansion coefficients via 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 calculation was previously described in our conference publication [8].

* Corresponding author.
E-mail addresses: erik.sausa@gmail.com (E. Šauša), pavel.rajmic@vut.cz (P. Rajmic), franz.hlawatsch@tuwien.ac.at (F. Hlawatsch).

https://doi.org/10.1016/j.sigpro.2023.109259
Received 6 February 2023; Received in revised form 12 September 2023; Accepted 13 September 2023
Available online 22 September 2023

- 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 localized character 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 time-dependent "region of interest". This region of interest is determined online and adaptively in a distributed manner.
- Efficient binary representations of the information communicated between the sensors.

The LC 2.0 method proposed in this article employs an appropriate combination of these innovations. We demonstrate experimentally that this can result in a reduction of intersensor communication by a factor of about 190, in addition to yielding a significant reduction of computational complexity. 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 2 describes our system model and reviews LC-based distributed particle filtering. Section 3 presents an OMP-based calculation of the expansion coefficients and a uniform-grid evaluation of the local log-likelihood functions. In Section 4, the B-spline dictionary is introduced. A distributed method for adaptively determining a region of interest is described in Section 5. Section 6 presents efficient binary coefficient representations. In Section 7, 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 8.

2. Review of LC-based distributed particle filtering

First, we review the conventional formulation of LC-based distributed particle filtering [4,5].

2.1. System model and local particle filter

We consider a target whose state $\mathbf{x}_n = (x_{n,1} \cdots x_{n,M})^{\mathrm{T}} \in \mathbb{R}^M$ (which, for example, includes position and velocity) evolves with time $n \in \mathbb{N}_0$ according to a known state-transition probability density function (pdf) $f(\mathbf{x}_n|\mathbf{x}_{n-1})$. There are S sensors indexed by $s \in \{1,\ldots,S\}$. Sensor s communicates with a certain set $\mathcal{N}_s \subseteq \{1,\ldots,S\}\setminus\{s\}$ 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 with multiple links – between any two sensors. At each time n, each sensor s acquires a measurement $\mathbf{z}_n^{(s)}$ (such as, e.g., noisy observations of the target's range and bearing) that is statistically related to the target state $\mathbf{x}_n$ according to the known *local likelihood function* (LLF) $f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$.

The *global likelihood function* (GLF) $f(\mathbf{z}_n|\mathbf{x}_n)$ involves the all-sensors measurement vector $\mathbf{z}_n \triangleq (\mathbf{z}_n^{(1)\mathrm{T}} \cdots \mathbf{z}_n^{(S)\mathrm{T}})^{\mathrm{T}}$, i.e., the vector stacking 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.,

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

Each sensor s knows its own measurement $\mathbf{z}_n^{(s)}$ and its own LLF $f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ (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, $\mathbf{z}_{1:n} \triangleq (\mathbf{z}_1^{\mathrm{T}} \cdots \mathbf{z}_n^{\mathrm{T}})^{\mathrm{T}}$. To this end, each sensor 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 2.2). The local particle filter at sensor s computes an approximation to the minimum mean-square error (MMSE) estimator [24] $\hat{\mathbf{x}}_n^{\mathrm{MMSE}} \triangleq \mathrm{E}\{\mathbf{x}_n|\mathbf{z}_{1:n}\} = \int_{\mathbb{R}^M} \mathbf{x}_n f(\mathbf{x}_n|\mathbf{z}_{1:n})\,\mathrm{d}\mathbf{x}_n$, in which $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ is the *global posterior pdf*. This global posterior pdf is represented in the local particle filter by J pairs of particles and associated weights, $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)}\big)\big\}_{j=1}^{J}$, with $\sum_{j=1}^{J} w_n^{(s,j)} = 1$.

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 $f\big(\mathbf{x}_n|\mathbf{x}_{n-1}^{(s,j)}\big)$, 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

$$w_{n|n-1}^{(s,j)} = c\, \hat{f}_s\big(\mathbf{z}_n|\mathbf{x}_{n|n-1}^{(s,j)}\big), \quad j = 1,\ldots,J, \tag{2}$$

with normalization factor $c = 1/\sum_{j=1}^{J} \hat{f}_s\big(\mathbf{z}_n|\mathbf{x}_{n|n-1}^{(s,j)}\big)$. 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, $\mathbf{z}_n$, and is calculated in a distributed way via the LC scheme reviewed in Section 2.2. 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 $\mathbf{x}_n^{(s,j)}$, $j = 1,\ldots,J$ with associated weights $w_n^{(s,j)} \equiv 1/J$. The overall recursive algorithm is initialized at time $n = 0$ by particles $\mathbf{x}_0^{(s,j)}$, $j = 1,\ldots,J$ 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 $\hat{\mathbf{x}}_n^{\mathrm{MMSE}}$ as the weighted sample mean of the predicted particles (before resampling), $\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^{J} w_{n|n-1}^{(s,j)} \mathbf{x}_{n|n-1}^{(s,j)}$.

2.2. 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 (2). Let us consider

$$L_n(\mathbf{x}_n) \triangleq \frac{1}{S}\log f(\mathbf{z}_n|\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S} \log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big), \tag{3}$$

where log denotes the natural logarithm and (1) was used. Note that, conversely, $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(SL_n(\mathbf{x}_n))$. 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.,

$$\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) \approx \sum_{k=1}^{K} \alpha_n^{(s,k)} \psi_k(\mathbf{x}_n). \tag{4}$$

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 3. The choice of the dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ will be considered in Section 4. By inserting (4) into (3), we obtain the following approximation of $L_n(\mathbf{x}_n) = \frac{1}{S}\log f(\mathbf{z}_n|\mathbf{x}_n)$:

$$L_n(\mathbf{x}_n) \approx \sum_{k=1}^{K} \beta_n^{(k)} \psi_k(\mathbf{x}_n), \tag{5}$$

with the *global* expansion coefficients $\beta_n^{(k)} \triangleq \frac{1}{S}\sum_{s=1}^{S} \alpha_n^{(s,k)}$, $k = 1,\ldots,K$.

The global expansion coefficients can be computed in a distributed manner by means of K instances of the average consensus algorithm [27]. 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

$$\hat{\beta}_n^{(k,s)}[i] = \sum_{s' \in \{s\}\cup\mathcal{N}_s} \gamma_{s,s'}\, \hat{\beta}_n^{(k,s')}[i-1]. \tag{6}$$

Here, the $\gamma_{s,s'}$ are suitably chosen weights, such as the Metropolis weights [27–29], and the $\hat{\beta}_n^{(k,s')}[i-1]$, $s' \in \mathcal{N}_s$ were communicated to sensor s by its neighboring sensors $s' \in \mathcal{N}_s$. Sensor s then broadcasts the updated iterated coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$, $k = 1, \dots, K$ to its neighboring sensors.

The recursion (6) is initialized by $\hat{\beta}_n^{(k,s)}[0] = \alpha_n^{(s,k)}$ and terminated after a sufficient number I of iterations. The final estimates $\hat{\beta}_n^{(k,s)}[I]$ are then used in (5) to obtain an approximation to $L_n(\mathbf{x}_n)$. Thus, the resulting approximation to $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(SL_n(\mathbf{x}_n))$ obtained at sensor s is

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

This is used in the update step (2). For $I \to \infty$, the consensus recursion would converge to $\beta_n^{(k)}$ because, as assumed in Section 2.1, 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 $\hat{\beta}_n^{(k,s)}[i]$, $k = 1, \dots, K$ to its neighbors $s' \in \mathcal{N}_s$. In the next four sections, we will propose modifications of the original LC scheme that lead to a significant reduction of the communication cost.

3. Sparsity-promoting LLF approximation using the OMP

In this section, we propose a sparsity-promoting method for calculating the local expansion coefficients $\{\alpha_n^{(s,k)}\}_{k=1}^{K}$ that are involved in the log-LLF approximation (4).

3.1. 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 (4) to the left hand side in a way such that the total approximation error at the predicted particles $\{\mathbf{x}_{n|n-1}^{(s,j)}\}_{j=1}^{J}$ is minimized. Let us introduce the local coefficient vector $\boldsymbol{\alpha}_n^{(s)} \triangleq (\alpha_n^{(s,1)} \cdots \alpha_n^{(s,K)})^{\mathrm{T}} \in \mathbb{R}^K$, the discretized-atom vector $\boldsymbol{\psi}_{n,k}^{(s)} \triangleq (\psi_k(\mathbf{x}_{n|n-1}^{(s,1)}) \cdots \psi_k(\mathbf{x}_{n|n-1}^{(s,J)}))^{\mathrm{T}} \in \mathbb{R}^J$, and the discretized-log-LLF vector $\boldsymbol{\eta}_n^{(s)} \triangleq (\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_{n|n-1}^{(s,1)}) \cdots \log f(\mathbf{z}_n^{(s)}|\mathbf{x}_{n|n-1}^{(s,J)}))^{\mathrm{T}} \in \mathbb{R}^J$. Then, the error minimized by the LS fit is given by

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

where the discretized-dictionary matrix $\boldsymbol{\Psi}_n^{(s)} \in \mathbb{R}^{J\times K}$ has the vectors $\boldsymbol{\psi}_{n,k}^{(s)}$, $k = 1, \dots, K$ as its 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 $\boldsymbol{\Psi}_n^{(s)+} \triangleq (\boldsymbol{\Psi}_n^{(s)\mathrm{T}} \boldsymbol{\Psi}_n^{(s)})^{-1} \boldsymbol{\Psi}_n^{(s)\mathrm{T}}$ 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.

3.2. 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 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.,

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

(Note that k_1 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 each further iteration $l = 2, 3, \dots$, 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.,

$$k_l = \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}} \frac{\left| \boldsymbol{\psi}_{n,k}^{(s)\mathrm{T}} \boldsymbol{\rho}_{n,l-1}^{(s)} \right|}{\|\boldsymbol{\psi}_{n,k}^{(s)}\|}, \qquad l = 2, 3, \dots .$$

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 l-dimensional subspace of $\mathbb{R}^J$ spanned by $\boldsymbol{\psi}_{n,k_1}^{(s)}, \dots, \boldsymbol{\psi}_{n,k_l}^{(s)}$, i.e.,

$$\boldsymbol{\rho}_{n,l}^{(s)} = \boldsymbol{\eta}_n^{(s)} - \mathbf{P}_{n,l}^{(s)} \boldsymbol{\eta}_n^{(s)}, \tag{7}$$

with the orthogonal projection matrix

$$\mathbf{P}_{n,l}^{(s)} \triangleq \boldsymbol{\Phi}_{n,l}^{(s)} (\boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}} \boldsymbol{\Phi}_{n,l}^{(s)})^{-1} \boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}} = \boldsymbol{\Phi}_{n,l}^{(s)} \boldsymbol{\Phi}_{n,l}^{(s)+}, \tag{8}$$

where $\boldsymbol{\Phi}_{n,l}^{(s)} \in \mathbb{R}^{J\times l}$ is the matrix with columns $\boldsymbol{\psi}_{n,k_1}^{(s)}, \dots, \boldsymbol{\psi}_{n,k_l}^{(s)}$. We note that (7), (8) 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 approximation of $\boldsymbol{\eta}_n^{(s)}$ by $\boldsymbol{\Phi}_{n,l}^{(s)}\mathbf{c}$, i.e., $\mathbf{c}_{n,l}^{(s)} = \operatorname{argmin}_{\mathbf{c}} \|\boldsymbol{\eta}_n^{(s)} - \boldsymbol{\Phi}_{n,l}^{(s)}\mathbf{c}\|^2$.

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 $L_s \leq K$ denote the final iteration index at sensor s. Then the result of the OMP algorithm is the coefficient vector $\boldsymbol{\alpha}_{n,\mathrm{OMP}}^{(s)} = (\alpha_{n,\mathrm{OMP}}^{(s,1)} \cdots \alpha_{n,\mathrm{OMP}}^{(s,K)})^{\mathrm{T}}$ 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 $k \in \{k_1, \dots, k_{L_s}\}$ and zero otherwise, i.e.,

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

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 can be 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 $l = 1, \dots, L_s$, whereas the LS-based calculation performs a single inversion of the matrix $\boldsymbol{\Psi}_n^{(s)\mathrm{T}} \boldsymbol{\Psi}_n^{(s)} \in \mathbb{R}^{K\times K}$. Note, however, that the matrices $\boldsymbol{\Phi}_{n,l}^{(s)\mathrm{T}} \boldsymbol{\Phi}_{n,l}^{(s)}$ are typically much smaller than the matrix $\boldsymbol{\Psi}_n^{(s)\mathrm{T}} \boldsymbol{\Psi}_n^{(s)}$.

The fact that a prescribed sparsity is obtained by terminating the OMP after a fixed number of iterations is an advantage of the proposed method. Other sparse approximation methods such as ℓ_1-based methods [32] usually need careful parameter tuning to achieve a solution with a prescribed sparsity.

3.3. Uniform sampling of the log-LLF

In both the OMP-based and LS-based calculation of the local expansion coefficients $\{\alpha_n^{(s,k)}\}_{k=1}^{K}$, the log-LLF $\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n)$ and the approximating function $\sum_{k=1}^{K} \alpha_n^{(s,k)} \psi_k(\mathbf{x}_n)$ (see (4)) are sampled at the predicted particles $\mathbf{x}_n = \mathbf{x}_{n|n-1}^{(s,j)}$, $j = 1, \dots, J$. However, when using a B-spline dictionary as proposed in Section 4.2, 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 coefficients are always affected by all the particles. To avoid this issue, we propose to use a *uniform* sampling of $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ and $\sum_{k=1}^{K} \alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$, where the samples $\mathbf{x}_n^{(q)} = \big(x_{n,1}^{(q)} \cdots x_{n,M}^{(q)}\big)^{\mathrm{T}}$, $q = 1,\dots,Q_n$ lie on a regular grid within a *region of interest* (ROI)

$$\mathcal{R}_n = \big[a_n^{(1)}, b_n^{(1)}\big] \times \cdots \times \big[a_n^{(M)}, b_n^{(M)}\big] \subset \mathbb{R}^M. \tag{9}$$

More specifically, in the mth coordinate direction, where $m \in \{1,\dots,M\}$, 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 5.

Using this uniform sampling approach, the atom vector $\boldsymbol{\psi}_{n,k}$ and the log-LLF vector $\boldsymbol{\eta}_n^{(s)}$ are redefined as $\boldsymbol{\psi}_{n,k} \triangleq \big(\psi_k(\mathbf{x}_n^{(1)}) \cdots \psi_k(\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}} \in \mathbb{R}^{Q_n}$ and $\boldsymbol{\eta}_n^{(s)} \triangleq \big(\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(1)}) \cdots \log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}} \in \mathbb{R}^{Q_n}$. We also obtain modified 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.

4. B-spline dictionary

The dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ used in the log-LLF approximation (4) 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.

4.1. 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 8. 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 $m \in \{1,\dots,M\}$, are defined as $\psi_{\tilde{k}}^{(m)}(x) = \cos\Big(\frac{2\pi}{d^{(m)}}(\tilde{k}-1)(x-a^{(m)})\Big)$ for $\tilde{k} = 1,\dots,\tilde{K}_m+1$ and $\psi_{\tilde{k}}^{(m)}(x) = \sin\Big(\frac{2\pi}{d^{(m)}}(\tilde{k}-1-\tilde{K}_m)(x-a^{(m)})\Big)$ for $\tilde{k} = \tilde{K}_m+2,\dots,2\tilde{K}_m+1$, 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

$$\tilde{\psi}_{\tilde{\mathbf{k}}}(\mathbf{x}) = \prod_{m=1}^{M} \psi_{\tilde{k}_m}^{(m)}(x_m), \tag{10}$$

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

4.2. 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]. 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, and especially desirable in our case in view of the localized character of the posterior distribution. Furthermore, B-splines are continuous and have continuous derivatives up to degree $r-1$. They are well suited for interpolation.

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]

$$\psi(x) = \begin{cases} \frac{2}{3} - |x|^2 + \frac{1}{2}|x|^3, & 0 \le |x| < 1 \\ \frac{1}{6}(2-|x|)^3, & 1 \le |x| < 2 \\ 0, & |x| \notin [0,2). \end{cases} \tag{11}$$

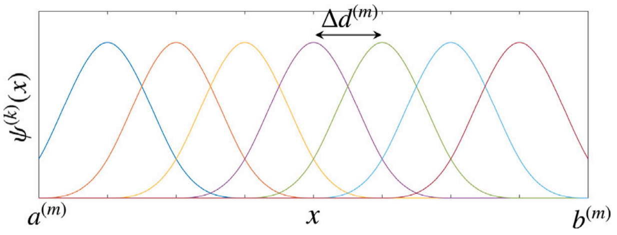


Fig. 1. 1-D B-spline dictionary as defined by (12), with $a^{(m)} = 0$, $b^{(m)} = 8$, $\tilde{K}_m = 7$, and $\Delta d^{(m)} = (b^{(m)} - a^{(m)})/(\tilde{K}_m + 1) = 1$.

The atoms of the 1-D B-spline dictionary for the mth coordinate direction, $\{\psi_{\tilde{k}}^{(m)}(x)\}_{\tilde{k}=1}^{\tilde{K}_m}$, are then defined by scaling and shifting the B-spline prototype $\psi(x)$ in (11) according to

$$\psi_{\tilde{k}}^{(m)}(x) = \psi\left(\frac{x - a^{(m)} - \tilde{k}\,\Delta d^{(m)}}{\Delta d^{(m)}}\right), \qquad x \in \big[a^{(m)}, b^{(m)}\big], \tag{12}$$

for $\tilde{k} = 1,\dots,\tilde{K}_m$. 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 $\psi_{\tilde{k}}^{(m)}(x)$ are centered around grid points $x_{\tilde{k}} = a^{(m)} + \tilde{k}\,\Delta d^{(m)}$, $\tilde{k} = 1,\dots,\tilde{K}_m$ that are placed uniformly in the interval $\big[a^{(m)}, b^{(m)}\big]$ with spacing $\Delta d^{(m)}$. An example is shown in Fig. 1.

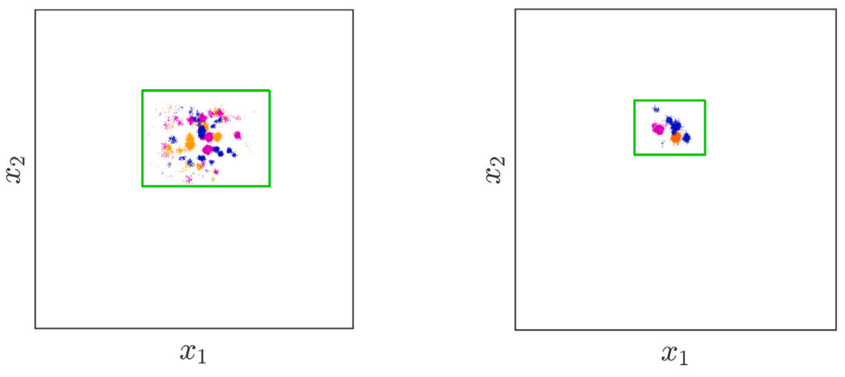
The M-D B-spline dictionary on the ROI $\mathcal{R}$ is then composed of M-D atoms $\tilde{\psi}_{\tilde{\mathbf{k}}}(\mathbf{x})$ that are constructed as in (10), with M-D index $\tilde{\mathbf{k}} = (\tilde{k}_1 \cdots \tilde{k}_M)^{\mathrm{T}}$ where $\tilde{k}_m \in \{1,\dots,\tilde{K}_m\}$. Finally, 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 $k \in \{1,\dots,K\}$, 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.

5. Distributed calculation of the ROI

According to Section 4, the choice of the ROI $\mathcal{R}_n$ in (9) influences the number of 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 $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ as described in Section 3.3, then $\mathcal{R}_n$ also influences the number Q_n of M-D samples $\mathbf{x}_n^{(1)},\dots,\mathbf{x}_n^{(Q_n)}$. Thus, the choice of $\mathcal{R}_n$ has an influence on the accuracy of the log-LLF approximation and 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.

5.1. Calculation of the ROI interval bounds

The proposed algorithm calculates the "ROI 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 "ROI extent vector" $\boldsymbol{d}_n = (d_n^{(1)} \cdots d_n^{(M)})^{\mathrm{T}}$ with $d_n^{(m)} = b_n^{(m)} - a_n^{(m)}$ at each time n in a distributed manner. This is 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. Note that the ROI interval bounds $a_n^{(m)}$ and $b_n^{(m)}$ can be recovered from $\boldsymbol{\xi}_n$ and $\boldsymbol{d}_n$ as $a_n^{(m)} = \xi_n^{(m)} - d_n^{(m)}/2$ and $b_n^{(m)} = \xi_n^{(m)} + d_n^{(m)}/2$.



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

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

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

$$\sigma^{(s)2}_{n,m} \triangleq \frac{1}{J}\sum_{j=1}^{J}\big(x^{(s,j)}_{n|n-1,m} - \hat{x}^{(s)}_{n,m}\big)^2, \quad m = 1, \dots, M,$$

where $\hat{x}^{(s)}_{n,m} \triangleq \frac{1}{J}\sum_{j=1}^{J} x^{(s,j)}_{n|n-1,m}$; here, $x^{(s,j)}_{n|n-1,m}$ denotes the mth element of $\mathbf{x}^{(s,j)}_{n|n-1}$. Then, for each m, $\hat{x}^{(s)}_{n,m}$ and $\sigma^{(s)2}_{n,m}$ are averaged across all sensors, i.e., the goal is to obtain $\bar{\hat{x}}_{n,m} \triangleq \frac{1}{S}\sum_{s=1}^{S}\hat{x}^{(s)}_{n,m}$ and $\overline{\sigma^2_{n,m}} \triangleq \frac{1}{S}\sum_{s=1}^{S}\sigma^{(s)2}_{n,m}$. We therefore perform $2M$ instances of the average consensus algorithm to calculate approximations $\bar{\hat{x}}^{(s)}_{n,m}$ of $\bar{\hat{x}}_{n,m}$ and $\overline{\sigma^{2(s)}_{n,m}}$ of $\overline{\sigma^2_{n,m}}$, for $m = 1, \dots, M$. After a finite number of consensus iterations, the approximations $\bar{\hat{x}}^{(s)}_{n,m}$ and $\overline{\sigma^{2(s)}_{n,m}}$ at different sensors s will be (slightly) different. Because the LC requires the same ROI $\mathcal{R}_n$ at each sensor, we next perform $2M$ instances of the maximum consensus algorithm [33] to calculate $\bar{\hat{x}}^{\max}_{n,m} \triangleq \max\{\bar{\hat{x}}^{(s)}_{n,m}\}_{s=1}^{S}$ and $\overline{\sigma^2}^{\max}_{n,m} \triangleq \max\{\overline{\sigma^{2(s)}_{n,m}}\}_{s=1}^{S}$ for $m = 1, \dots, M$. The maximum consensus algorithm converges in a finite number of iterations, which equals the diameter of the sensor network [33]. Hence, after performing these iterations, identical maxima $\bar{\hat{x}}^{\max}_{n,m}$ and $\overline{\sigma^2}^{\max}_{n,m}$ are available at all sensors, as desired.

We then define the ROI center point ξ_n to be the vector with elements

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

Next, we choose each extent vector element $d^{(m)}_n$ as a function of the standard deviation $s_{n,m} \triangleq \sqrt{\overline{\sigma^2}^{\max}_{n,m}}$. The $d^{(m)}_n$ should be 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 $d^{(m)}_n$ equal to $\gamma s_{n,m}$, where $\gamma > 1$ is an empirically chosen scaling factor, subject to a lower bound $d^{(m)}_{\min}$, i.e.,

$$d^{(m)}_n = \begin{cases} \gamma s_{n,m}, & \text{if } \gamma s_{n,m} \ge d^{(m)}_{\min}, \\ d^{(m)}_{\min}, & \text{if } \gamma s_{n,m} < d^{(m)}_{\min}. \end{cases} \tag{13}$$

Here, the lower bound $d^{(m)}_{\min}$ adds robustness in cases where the set of predicted particles $\{\mathbf{x}^{(s,j)}_{n|n-1}\}_{j=1}^{J}$ is highly concentrated in the state space. Fig. 2 shows an example of the ROI $\mathcal{R}_n$ at two successive time points n along with the predicted particles.

5.2. Calculation of the numbers of atoms

Once the extent parameters $d^{(m)}_n$ are known, we are also able to calculate the numbers of 1-D B-spline or Fourier atoms $\psi^{(m)}_{\tilde{k}}(x)$. We recall from Section 4.2 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 $\Delta d^{(m)}_n = d^{(m)}_n/(\tilde{K}_{n,m}+1)$, respectively. Therefore, for a specified B-spline spacing $\Delta d^{(m)}_n$, we obtain

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

where $\lceil x \rceil$ denotes the smallest integer not smaller than x. 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, $\kappa_{n,m} \triangleq \tilde{K}_{n,m}/d^{(m)}_n$. Here, $\tilde{K}_{n,m}$ follows as

$$\tilde{K}_{n,m} = \big\lceil \kappa_{n,m} d^{(m)}_n \big\rceil. \tag{14}$$

For the Fourier dictionary, according to Section 4.1, 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 $\nu_{\tilde{k}} \triangleq (\tilde{k}-1)/d^{(m)}_n$, $\tilde{k} = 1, \dots, \tilde{K}_m + 1$. The bandwidth is given by the maximum frequency, $\nu_{\tilde{K}_{n,m}+1} = \tilde{K}_{n,m}/d^{(m)}_n$. If $\nu_{\tilde{K}_{n,m}+1}$ is specified, $\tilde{K}_{n,m}$ is obtained as

$$\tilde{K}_{n,m} = \big\lfloor \nu_{\tilde{K}_{n,m}+1} d^{(m)}_n \big\rfloor,$$

where $\lfloor x \rfloor$ denotes the largest integer not larger than x.

The overall dictionary size, i.e., the number of M-D atoms $\psi_k(\mathbf{x})$, $k = 1, \dots, K_n$, 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 $\log f(\mathbf{z}^{(s)}_n|\mathbf{x}_n)$ within $\mathcal{R}_n$ but also implies a larger dictionary size K_n and, potentially, a higher communication cost.

6. Binary coefficient representation

According to Section 2.2, in LC iteration i, each sensor s broadcasts its iterated coefficient estimates $\hat{\beta}^{(k,s)}_n[i]$, $k = 1, \dots, K_n$ to the neighboring sensors $s' \in \mathcal{N}_s$. In practice, this is done in a binary format. If each $\hat{\beta}^{(k,s)}_n[i]$ is represented by a bit sequence of length n_{b}, then $N_{\mathrm{naive}} = K_n n_{\mathrm{b}}$ bits have to be broadcast by sensor s in LC iteration i. However, this communication cost can be significantly reduced by broadcasting only the *nonzero* $\hat{\beta}^{(k,s)}_n[i]$ 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 $\hat{\beta}^{(k,s)}_n[i]$. Before the first consensus iteration $i = 1$, the coefficient estimates are initialized as $\hat{\beta}^{(k,s)}_n[0] = \alpha^{(s,k)}_n$, and thus $L_s[0] = L_s$, where L_s denotes the number of nonzero $\alpha^{(s,k)}_n$ at sensor s. If the OMP method was used for calculating the $\alpha^{(s,k)}_n$, then, according to Section 3.2, L_s is given by the number of OMP iterations. If the LS

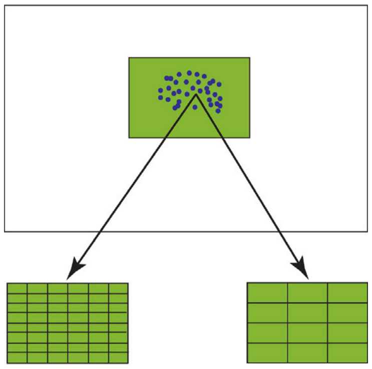


Fig. 3. Illustration of the covering of the ROI $\mathcal{R}_n$ (green rectangle) with B-spline atoms of two different densities, for dimension $M = 2$. Also shown are the predicted particles $\mathbf{x}_{n|n-1}^{(s,j)}$ for all the sensors (blue dots). The small grid rectangles within the ROI indicate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

method was used, then, according to Section 3.1, the number of $\alpha_n^{(s,k)}$ is nominally equal to the total number of dictionary atoms at time n, K_n, but 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 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 (6) will produce some additional nonzero $\hat{\beta}_n^{(k,s)}[i]$. However, $L_s[i]$ will still be much smaller than K_n.

In the following, we propose 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

$$N_{\mathrm{indicator},i}^{(s)} = L_s[i]\, n_\mathrm{b} + K_n \,. \tag{15}$$

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

$$N_{\mathrm{label},i}^{(s)} = L_s[i] \left(n_\mathrm{b} + \lceil \log_2(K_n) \rceil \right). \tag{16}$$

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 $L_s[i] < K_n / \lceil \log_2(K_n) \rceil$.

Finally, the *hyperrectangle method* is specifically suited to the B-spline dictionary. As described in Section 4.2, 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 $\tilde{\mathbf{k}} = (\tilde{k}_1 \cdots \tilde{k}_M)^\mathrm{T}$ with $\tilde{k}_m \in \{1, \ldots, \tilde{K}_m\}$ (see (12)). 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 small subregion of $\mathcal{R}_n$. Equivalently, the associated M-D indices $\tilde{\mathbf{k}}$ are located in a small M-D "discrete hyperrectangle" $\mathcal{K}_n^{(s)}[i]$ that consists of all $\tilde{\mathbf{k}}$ with $\tilde{k}_m \in \big\{\tilde{l}_{n,m}^{(s)}[i], \ldots, \tilde{l}_{n,m}^{(s)}[i] + \Delta\tilde{k}_{n,m}^{(s)}[i]\big\}$ 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.

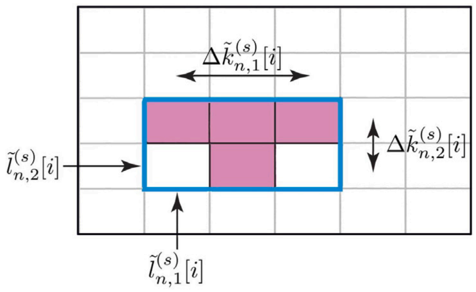

$\Delta\tilde{k}_{n,1}^{(s)}[i]$
$\Delta\tilde{k}_{n,2}^{(s)}[i]$
$\tilde{l}_{n,2}^{(s)}[i]$
$\tilde{l}_{n,1}^{(s)}[i]$

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 magenta 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 one of $K_n = \prod_{m=1}^{M} \tilde{K}_{n,m}$ 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 $\boldsymbol{\Delta}_n^{(s)}[i]$. The total number of bits broadcast by sensor s in LC iteration i is thus $N_{\mathrm{hyperrect},i}^{(s)} = L_s[i] n_\mathrm{b} + |\mathcal{K}_n^{(s)}[i]| + \lceil \log_2(K_n) \rceil + \lceil \log_2(N_\Delta) \rceil$ 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 (15) and (16), 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.

7. LC 2.0 for the distributed PDAF

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).

7.1. Measurement model, LLF, and GLF

The measurement model underlying the PDAF extends the DPF measurement model of Section 2.1 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 $f_\mathrm{t}^{(s)}\big(\mathbf{z}_{n,w}^{(s)} | \mathbf{x}_n\big)$. (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

$$f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) = l_{n,0}^{(s)} + \sum_{w=1}^{W_n^{(s)}} l_{n,w}^{(s)}(\mathbf{x}_n), \tag{17}$$

with the nonnegative constant "floor" component

$$l_{n,0}^{(s)} \triangleq C\big(\mathbf{z}_n^{(s)}\big)\big(1 - P_{\mathrm{d}}^{(s)}\big)\,\mu^{(s)} \tag{18}$$

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

$$l_{n,w}^{(s)}(\mathbf{x}_n) \triangleq C\big(\mathbf{z}_n^{(s)}\big)\, P_{\mathrm{d}}^{(s)} \frac{f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}|\mathbf{x}_n\big)}{f_{\mathrm{c}}^{(s)}\big(\mathbf{z}_{n,w}^{(s)}\big)}, \quad w = 1,\dots,W_n^{(s)}. \tag{19}$$

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 (17)–(19) would simplify to $f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) = f_{\mathrm{t}}^{(s)}\big(\mathbf{z}_{n,1}^{(s)}|\mathbf{x}_n\big)$, i.e., the floor component would be zero and there would be only one measurement-related component, which is target-generated; this equals the DPF measurement model of Section 2.1.

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 (17)–(19) [2,7]. For a distributed implementation, each sensor s runs a local particle filter as described in Section 2.1. 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 (2). However, a difference from the GLF calculation performed for the DPF is that now we also have to take into account the constant floor component $l_{n,0}^{(s)}$ of the LLF (17).

7.2. Splitting off the floor

We now propose a modification of the LC that promotes a sparse representation of the LLF (17) 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 (17), the LLF $f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ 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 $f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$, i.e., we effectively have $l_{n,0}^{(s)} = \min_{\mathbf{x}_n\in\mathbb{R}^M} f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$, and consequently

$$f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) \geq l_{n,0}^{(s)}. \tag{20}$$

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

$$\lambda_{n,0}^{(s)} \triangleq \min_{\mathbf{x}_n\in\mathbb{R}^M} \log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big). \tag{21}$$

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

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

We conclude from (20) that $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) \geq \log l_{n,0}^{(s)}$ or, equivalently, using (22),

$$\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) \geq \lambda_{n,0}^{(s)}. \tag{23}$$

It follows from (23) that the log-LLF can be written as

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

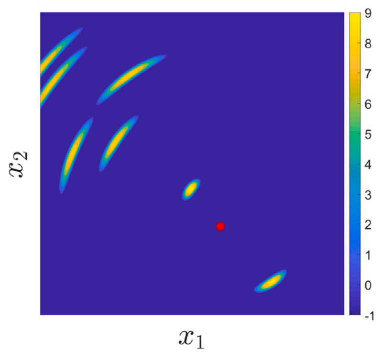


Fig. 5. Example (simulation result, using one sensor at the position indicated by the red bullet) of a log-LLF for dimension $M = 2$. This log-LLF comprises seven measurement-related components, whereof one is target-generated and the remaining six are clutter. The dark blue background corresponds to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

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 (24) into (3) yields

$$L_n(\mathbf{x}_n) = \frac{1}{S}\sum_{s=1}^{S}\big(\lambda_{n,0}^{(s)} + \lambda_n^{(s)}(\mathbf{x}_n)\big) = \bar{\lambda}_{n,0} + \bar{\lambda}_n(\mathbf{x}_n), \tag{25}$$

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

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

According to Section 2.2, 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 (26)) 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 overall 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 $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$. (In principle, the minimum can be obtained from the closed-form expression (18); however, we observed that a numerical computation according to (21) 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 $\lambda_n^{(s)}(\mathbf{x}_n) = \log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big) - \lambda_{n,0}^{(s)}$ (see (24)). 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 (25), i.e., by adding its local estimates of $\bar{\lambda}_{n,0}$ and $\bar{\lambda}_n(\mathbf{x}_n)$.

7.3. 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 adaptive 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 5.1 $\mathcal{R}_n$ 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.

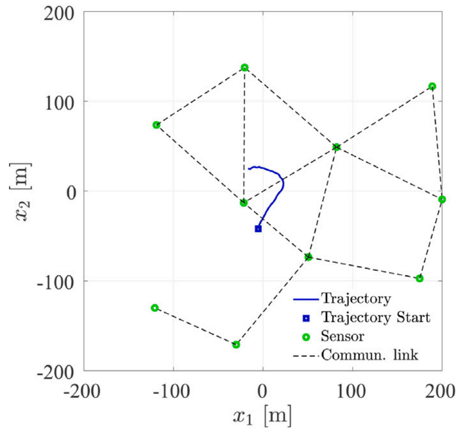


Fig. 6. Surveillance area, sensor network, and target trajectory used in our simulation.

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 (22), the log-LLF floor is now determined only on $\mathcal{R}_n$, i.e., $\tilde{\lambda}^{(s)}_{n,0} \triangleq \log\big(\min_{\mathbf{x}_n \in \mathcal{R}_n} f\big(\mathbf{z}^{(s)}_n|\mathbf{x}_n\big)\big)$, and the calculation of the floorless log-LLF $\tilde{\lambda}^{(s)}_n(\mathbf{x}_n) \triangleq \log f\big(\mathbf{z}^{(s)}_n|\mathbf{x}_n\big) - \tilde{\lambda}^{(s)}_{n,0}$ and the B-spline expansion of $\tilde{\lambda}^{(s)}_n(\mathbf{x}_n)$ are performed on $\mathcal{R}_n$.

8. Simulation results

In this section, we evaluate and compare the performance, communication cost, and computational complexity of LC 2.0 relative to the conventional LC. For a detailed evaluation 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.

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 target's position and velocity components, respectively. The surveillance area is $[-200\,\mathrm{m}, 200\,\mathrm{m}] \times [-200\,\mathrm{m}, 200\,\mathrm{m}]$. The evolution of $\mathbf{x}_n$ is modeled as $\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \boldsymbol{\Gamma}\mathbf{u}_n$, where the matrices $\mathbf{F} \in \mathbb{R}^{4\times 4}$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{4\times 2}$ are defined in [34], involving a time step parameter T that is chosen as 1 s, 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 $\sigma_u = 1/3\ \mathrm{m/s^2}$. 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 $\sigma_u^2\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\mathrm{T}}$. The sensor network consists of $S = 10$ sensors. Fig. 6 shows the surveillance area, the sensor network, and the target trajectory used in our simulation.

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

$$\mathbf{z}^{(s)}_n = \big(\|\tilde{\mathbf{x}}_n - \mathbf{p}^{(s)}\|\ \ \varphi(\tilde{\mathbf{x}}_n, \mathbf{p}^{(s)})\big)^{\mathrm{T}} + \mathbf{v}^{(s)}_n, \tag{27}$$

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 $\mathbf{v}^{(s)}_n$ 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 $\mathbf{z}^{(s)}_n$ depends only on the position $\tilde{\mathbf{x}}_n$, the LLF is effectively given by $f\big(\mathbf{z}^{(s)}_n|\tilde{\mathbf{x}}_n\big)$, which implies that our effective state-space dimension is $M = 2$. A modified measurement model will be used in Section 8.7.

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 and standard deviation 1/3 m/s, and the target heading is sampled from a uniform distribution on $(-180°, 180°]$. The LC employs $I = 20$ consensus iterations. 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 after the initial period $n = 1, \ldots, 10$ (i.e., after effects due to particle filter initialization have died off), the localization RMSE is smaller than $t_{\mathrm{loss}} = 5\,\mathrm{m}$; otherwise it is considered a track loss and included in the calculation of the track loss percentage. We note that the somewhat arbitrary threshold of 5 m was motivated by the fact that it is roughly 1% of the lateral length of the surveillance area (which is 400 m).

8.1. OMP

First, we demonstrate the savings in communication that are achieved by the OMP-based calculation of the local expansion coefficients described in Section 3.2 relative to the conventional LS-based calculation. We only consider a B-spline dictionary because the benefits of the OMP for a Fourier dictionary were already discussed in [8]. Our B-spline dictionary comprises $K = 400$ atoms located uniformly in the entire surveillance area. We used uniform sampling of the log-LLF on the entire surveillance area (see Section 3.3). That is, at this point, we do not use the adaptive ROI. Fig. 7(a) shows the time-averaged localization RMSE (averaged over all time steps $n = 1, \ldots, 50$) versus the number of nonzero local expansion coefficients. Here, as well as in the simulations reported later, L_s was chosen identically for each sensor s, i.e., $L_s \equiv L$. In the OMP case, L equals the number of OMP iterations while in the LS case, we retained the L local coefficients with the largest absolute values. In what follows, we consider $L \in \{2, \ldots, 20\}$. We conclude from Fig. 7(a) that for $L \geq 4$, OMP leads to a smaller (better) localization RMSE than LS. Conversely, using OMP, a given RMSE is obtained with a smaller L than using LS. For example, an RMSE of about 3.3 m is obtained with $L = 15$ when using LS as opposed to only $L = 6$ when using OMP. This superior RMSE performance of OMP can be explained by OMP's superior approximation accuracy, which is related to the fact that OMP performs several (small-size) LS steps and chooses the best coefficients one-by-one, whereas in the LS method the relevant coefficients are selected via a single heuristic thresholding that is performed after the LS approximation. To complete the picture, we note that neither OMP nor LS produced any track losses.

The smaller values of L required by the OMP translate into savings in the number of nonzero coefficient estimates $\hat{\beta}^{(k,s)}_n[i]$. Fig. 7(b) 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 has to broadcast for an RMSE of 3.3 m is reduced by about one half if OMP is used instead of LS. Because of this superiority of OMP, we will not further consider LS unless stated otherwise.

8.2. 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 $\tilde{K}_m = 20$ nonzero 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 8.5.) Both DPF-B and DPF-F use the OMP with $L = 5$ iterations.

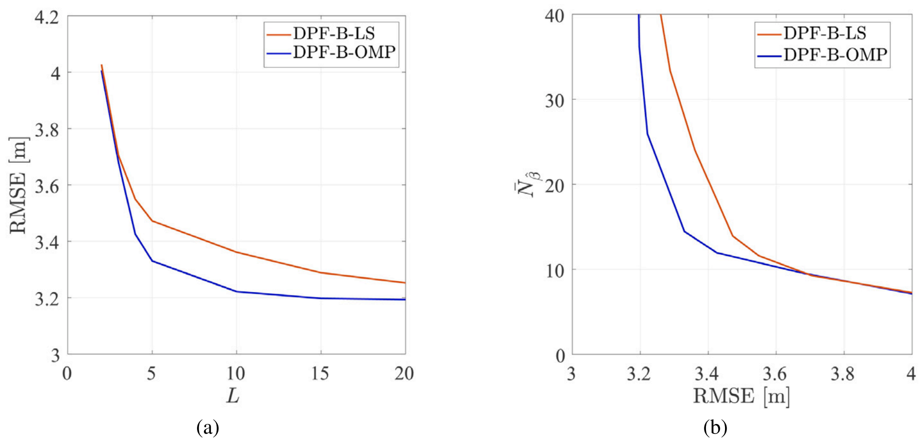


Fig. 7. Comparison of DPFs with OMP-based and LS-based calculation of the local expansion coefficients, using a B-spline dictionary and uniform sampling in the entire surveillance area (these DPFs are abbreviated as DPF-B-OMP and DPF-B-LS, respectively): (a) Time-averaged localization RMSE versus number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

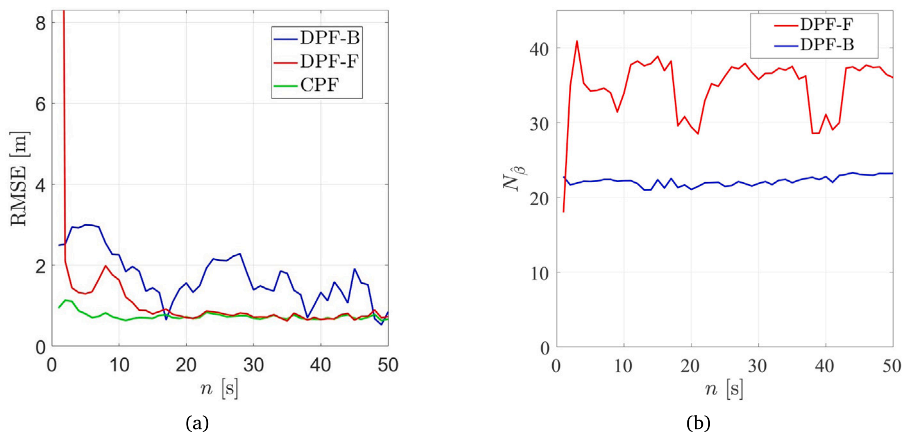


Fig. 8. Comparison of DPFs using a B-spline dictionary and a Fourier dictionary, both covering the entire surveillance area: (a) Localization RMSE versus time, (b) average number of nonzero coefficient estimates, $N_{\hat{\beta}}$, versus time.

Fig. 8(a) 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 (worse) 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 $n \geq 13$. The RMSE of DPF-B exhibits a large variation with time; in particular, it is large around $n = 25$, where the target changes direction (see Fig. 6). The higher RMSE and the RMSE's time variation can be explained by the small density of the B-spline grid. These issues will be resolved in Section 8.5 through the use of the adaptive ROI.

On the other hand, 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. 8(b) 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 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 8.5. Furthermore, $N_{\hat{\beta}}$ is seen to be fairly constant in the case of DPF-B, which implies an approximately constant communication cost. By contrast, in the case of DPF-F, $N_{\hat{\beta}}$ exhibits a large time variation. This can be explained by the fact, observed in our simulations, that using the Fourier dictionary the atoms selected by the OMP algorithm are partly different across the sensors for some time steps, whereas using the B-spline dictionary the atoms tend to coincide across the sensors for most time steps.

8.3. Binary representation

Our next simulation compares DPF-B variants using the binary coding methods presented in Section 6. 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 8.5 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 communication cost of this DPF-F variant is about twice that

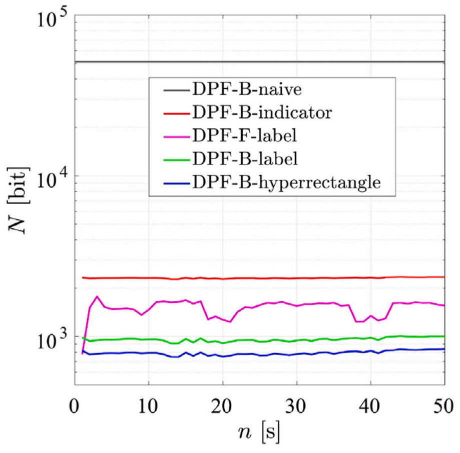


Fig. 9. Average binary communication costs of DPF-B variants using different binary coding methods, as well as of DPF-F using the label method.

Table 1
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

of DPF-B using the hyperrectangle method. Thus, using the B-spline dictionary instead of the Fourier dictionary reduces the communication cost by approximately one half.

8.4. Uniform log-LLF sampling

As mentioned in Section 3.3, 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 1 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). This can partially be attributed to particle degeneracy, and we conjecture that using a more sophisticated resampling filter would result in a smaller track loss percentage at the cost of a higher complexity. By contrast, uniform sampling yields excellent performance, with $\rho = 0.2\%$ or 0% and RMSE around $1.7\,\text{m}$ for $Q = 10000$ or larger.

8.5. Adaptive ROI

Next, we demonstrate the benefits of using the adaptive ROI introduced in Section 5.1. 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 (13)) 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 (14) with $\kappa_{n,1} = \kappa_{n,2} = 1/20$. 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 $Q_n = d_n^{(1)} d_n^{(2)}$, where the ROI interval lengths $d_n^{(1)}$ and $d_n^{(2)}$ are determined adaptively according to (13). For initialization of the local particle filters at $n = 1$, $\mathcal{R}_1$ is chosen as the entire surveillance area; furthermore, $\tilde{K}_{1,1} = \tilde{K}_{1,2} = 10$, corresponding to $K_1 = 100$ B-spline atoms. For $n = 1$, the sampling density is reduced to one sample per 5 m, therefore $Q_1 = 6400$. 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 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 $L = \min\{5, K_n\}$ iterations (note that L 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 $n_\text{b} = 32$. One can see in Fig. 10(a) that the localization RMSE of DPF-B-ROI is typically significantly lower than that of DPF-B. Moreover, for n larger than about 10, 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. 10(b) 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 8.2, which are again visible in Fig. 10(b).

Although our focus is on reducing the communication cost, also the computational complexity of the proposed DPF methodology is of interest. Fig. 10(c) shows the runtime per time step n, averaged over all sensors and simulation runs, of DPF-B-ROI and DPF-F versus time. These results were obtained with a MATLAB implementation on a laptop computer using 16 GB RAM and an i7-8565U CPU operating at a clock rate of 1.80 GHz. It is seen that the runtime of DPF-B-ROI, after a brief initial convergence phase, is about $0.01\,\text{s}$, which is only about one tenth of the runtime of DPF-F. This can be explained by the reduced dictionary size K_n and the reduced number of sampling points Q_n employed by DPF-B-ROI relative to DPF-F. The runtime of DPF-B-ROI can be broken down as follows: 62% is spent on the update step, 16% on calculating the expansion coefficients, 12% on particle resampling, 8% on the prediction step, and 2% on calculating the ROI parameters, such as the particles' standard deviations $s_{n,m}$. In DPF-F, about 50% of the runtime is spent on calculating the expansion coefficients. This large percentage is due to the fact that DPF-F uses sampling of the log-LLF at the particles, i.e., the Fourier basis functions have to be evaluated at the J particles, whereas in DPF-B-ROI only Q_n sampling points are used. As a consequence, in DPF-F, the matrices involved in the OMP are larger, resulting in a higher complexity. We report for completeness that the average runtime of DPF-B (i.e., not using an adaptive ROI) is approximately 3.4 s and thus significantly larger than that of DPF-B-ROI. Finally, we note that the complexity of evaluating the B-spline basis functions can be reduced by a table lookup based on a piecewise linear approximation, as discussed in [35].

8.6. 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. In addition, we compare the performance of LC and LC 2.0 with that of two diffusion-based methods. Fig. 11(a) shows the RMSE of DPF-B-ROI (previously considered in Fig. 10, but now dubbed DPF-LC2.0), of a DPF using the conventional LC (dubbed DPF-LC), and of the diffusion-based DPFs

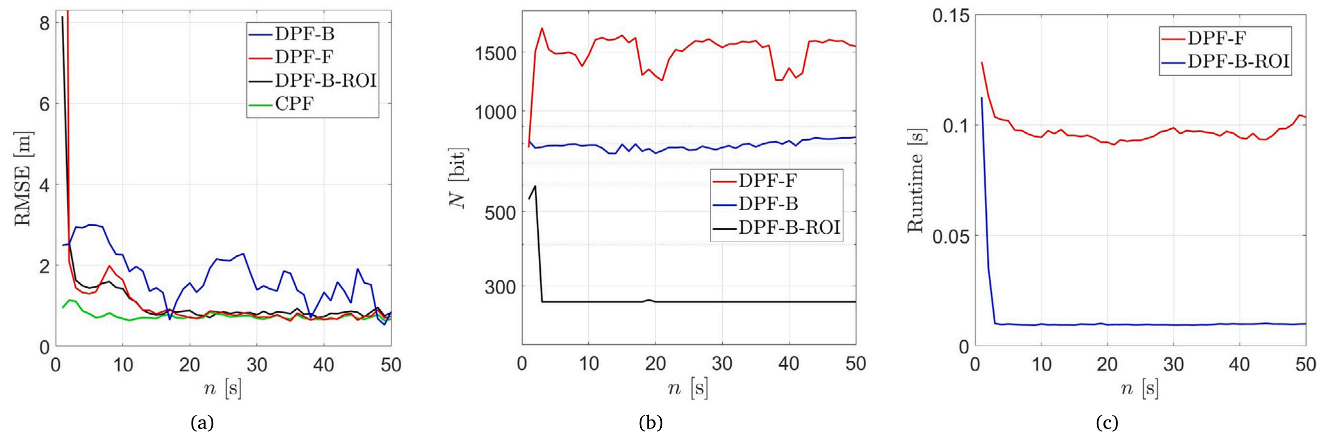


Fig. 10. Comparison of DPF-B with and without adaptive ROI: (a) Localization RMSE, (b) average binary communication cost, (c) average runtime per time step.

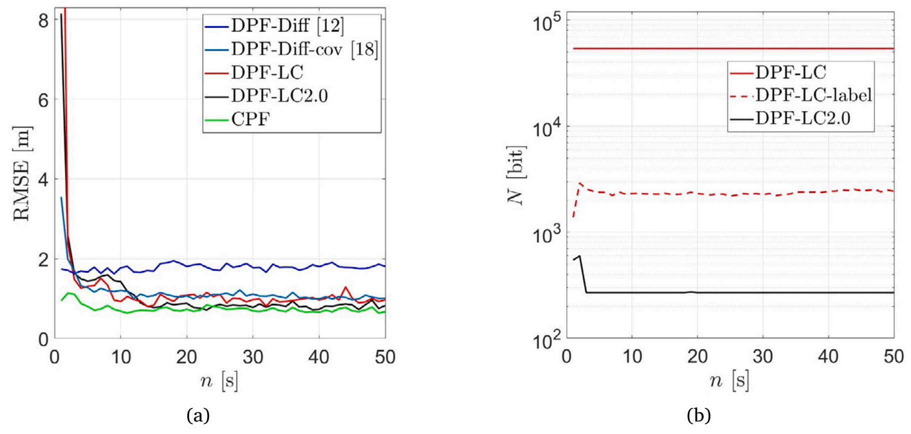


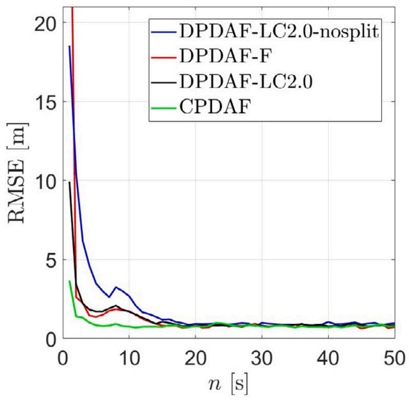
Fig. 11. Comparison of a DPF using LC2.0 (identical to DPF-B-ROI in Fig. 10) with a DPF using the conventional LC: (a) Localization RMSE, also showing the results for CPF and for the diffusion-based DPFs proposed in [12] and [17], (b) average binary communication cost, also showing the results for a DPF using the conventional LC enhanced by the label method for binary coding.

proposed in [12] (dubbed DPF-Diff) and in [17] (dubbed DPF-Diff-cov). 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 8.5. By contrast, DPF-LC uses LS-based calculation of the local expansion coefficients, retaining the $L = 10$ 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, we chose $L = 10$ and $K = 1681$ because this resulted in a similar localization RMSE as for DPF-LC2.0. One can see in Fig. 11(a) that, as intended, the localization RMSE of DPF-LC2.0 is similar to that of DPF-LC, and it closely approaches that of CPF. 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. It can furthermore be seen in Fig. 11(a) that, after an initial convergence phase, the localization RMSE of DPF-Diff and DPF-Diff-cov is, respectively, about 80% and 25% higher than that of DPF-LC2.0 and DPF-LC. This is consistent with the fact that the diffusion protocol does not aim at approximating the Bayes-optimal filter. The track loss percentage of DPF-Diff and DPF-Diff-cov was measured as 0.3% and 0.2%, respectively, which is similar to that of DPF-LC2.0.

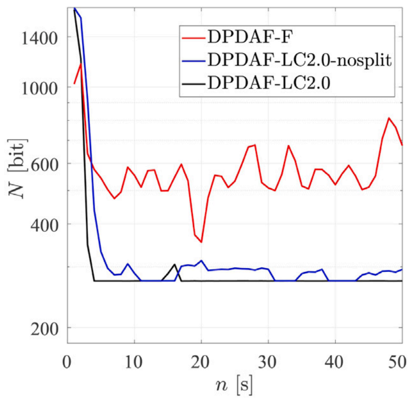
Fig. 11(b) shows that the communication cost of DPF-LC2.0 is only about 0.5% of that of DPF-LC. A large part of this reduction is caused by the efficient binary coding. To demonstrate this fact, Fig. 11(b) also considers a DPF-LC variant, designated DPF-LC-label, that uses the label coding method instead 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 $N = 284.14$ bit for DPF-LC2.0. Thus, relative to DPF-LC, efficient binary coding reduces communication by a factor of about 24, 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), resulting in a total reduction by a factor of about 190. Finally, the communication cost of DPF-Diff and of DPF-Diff-cov was measured as $N = 192$ bit and $N = 512$ bit, respectively. Note that this is the total communication cost, whereas the values of N given for the three consensus-based DPF variants are the communication cost per LC iteration. To obtain a meaningful comparison, we have to multiply the latter by the number of LC iterations, $I = 20$; we then see that the total communication cost of DPF-Diff and of DPF-Diff-cov is smaller by a factor of, respectively, about 25 and 11 than the total communication cost of DPF-LC2.0. Note, however, that the communication cost of DPF-Diff and DPF-Diff-cov increases linearly with the dimension of the measurements, whereas that of DPF-LC2.0 does not depend on the measurement dimension.

8.7. LC2.0 for the DPDAF

Finally, we consider the use of LC2.0 within a distributed PDAF (DPDAF) as proposed in Section 7. The simulation setup is as before, except that now there are multiple measurements per sensor with the following parameters (cf. Section 7.1): 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



(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.

surveillance area. 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 density $\kappa_{n,1} = \kappa_{n,2} = 1/20$, and initialized as $K_1 = 225$), and it splits off the log-GLF floor within the ROI as described in Section 7.3. Both DPDAF-LC2.0 and DPDAF-F use $L = \min\{5, K_n\}$ 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. 12(a) 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 15, it is similar. The track loss percentage was measured as $\rho = 0.4\%$ for both DPDAF-LC2.0 and DPDAF-F. For $n \leq 15$, the RMSE of DPDAF-LC2.0-nosplit is higher than that of DPDAF-LC2.0 and DPDAF-F. Moreover, the track loss percentage of DPDAF-LC2.0-nosplit was measured as 4.5% and is thus significantly higher than that of DPDAF-LC2.0. Hence, splitting off the log-GLF floor improves the tracking accuracy. Fig. 12(b) 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 $n \leq 10$. 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.

9. 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 reductions in communication by a factor of about 190, without a loss in tracking performance.

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

CRediT authorship contribution statement

E. Šauša: Software, Simulation, Visualization, Writing (draft and review). **P. Rajmic:** Conceptualization, Methodology, Writing (draft and review). **F. Hlawatsch:** Conceptualization, Methodology, Writing (draft and review).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

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 im Breisgau, Germany, June 2018. 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, N. Gordon, Beyond the Kalman Filter : Particle Filters for Tracking Applications, Artech House, Boston, MA, USA, 2004.
[2] S. Challa, M.R. Morelande, D. Mušicki, R.J. Evans, Fundamentals of Object Tracking, Cambridge University Press, New York, NY, USA, 2011.
[3] F. Zhao, L. Guibas, Wireless Sensor Networks: An Information Processing Approach, Morgan Kaufmann, San Francisco, CA, USA, 2004.
[4] O. Hlinka, O. Slučiak, F. Hlawatsch, P.M. Djurić, M. Rupp, Likelihood consensus and its application to distributed particle filtering, IEEE Trans. Signal Process. 60 (8) (2012) 4334–4349.
[5] O. Hlinka, F. Hlawatsch, P.M. Djurić, Consensus-based distributed particle filtering with distributed proposal adaptation, IEEE Trans. Signal Process. 62 (12) (2014) 3029–3041.
[6] Y. Bar-Shalom, F. Daum, J. Huang, The probabilistic data association filter, IEEE Control Syst. Mag. 29 (6) (2009) 82–100.
[7] Y. Bar-Shalom, P.K. Willett, X. Tian, Tracking and Data Fusion: A HandBook of Algorithms, Yaakov Bar-Shalom, Storrs, CT, USA, 2011.
[8] R. Repp, P. Rajmic, F. Meyer, F. Hlawatsch, Target tracking using a distributed particle-PDA filter with sparsity-promoting likelihood consensus, in: Proc. IEEE SSP, Freiburg im Breisgau, Germany, 2018, pp. 653–657.
[9] M.J. Coates, Distributed particle filters for sensor networks, in: Proc. IEEE/ACM IPSN-04, Berkeley, CA, USA, 2004, pp. 99–107.

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

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

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

[13] W. Xia, M. Sun, Z. Zhou, Diffusion collaborative feedback particle filter, IEEE Signal Process. Lett. 27 (2020) 1185–1189.

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

[15] X. Zhong, 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. 60 (9) (2012) 4719–4733.

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

[17] M.G. Bruno, S.S. Dias, A Bayesian interpretation of distributed diffusion filtering algorithms [lecture notes], IEEE Signal Process. Mag. 35 (3) (2018) 118–123.

[18] O. Hlinka, F. Hlawatsch, P.M. Djurić, Distributed particle filtering in agent networks: A survey, classification, and comparison, IEEE Signal Process. Mag. 30 (1) (2013) 61–81.

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

[20] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.

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

[22] M. Unser, Splines: A perfect fit for signal and image processing, IEEE Signal Process. Mag. 16 (6) (1999) 22–38.

[23] G. Papa, R. Repp, F. Meyer, P. Braca, F. Hlawatsch, Distributed Bernoulli filtering using likelihood consensus, IEEE Trans. Signal Inform. Process. Netw. 5 (2) (2019) 218–233.

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

[25] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2) (2002) 174–188.

[26] T. Li, M. Bolic, P.M. Djurić, Resampling methods for particle filtering: Classification, implementation, and strategies, IEEE Signal Process. Mag. 32 (3) (2015) 70–86.

[27] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (1) (2007) 215–233.

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

[29] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, Systems Control Lett. 53 (1) (2004) 65–78.

[30] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, A. Scaglione, Gossip algorithms for distributed signal processing, Proc. IEEE 98 (11) (2010) 1847–1864.

[31] Å. Björck, Least squares methods, in: P.G. Ciarlet, J.L. Lions (Eds.), HandBook of Numerical Analysis, Vol. 1, Elsevier, 1990, pp. 465–652.

[32] D.L. Donoho, M. Elad, Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization, Proc. Natl. Acad. Sci. USA 100 (5) (2003) 2197–2202.

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

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

[35] J. Detrey, F. de Dinechin, Table-Based Polynomials for Fast Hardware Function Evaluation, Research Report LIP RR-2004-52, Laboratoire de l'informatique du parallélisme, 2004.