LIKELIHOOD CONSENSUS 2.0: REDUCING INTERAGENT COMMUNICATION IN DISTRIBUTED BAYESIAN TARGET TRACKING

Erik Šauša⋆, Pavel Rajmic†, and Franz Hlawatsch⋆

⋆Institute of Telecommunications, TU Wien, Vienna, Austria
†Department of Telecommunications, Brno University of Technology, Brno, Czech Republic

ABSTRACT

We propose a communication-efficient scheme for distributed Bayesian target tracking (distributed particle filtering) in possibly nonlinear and non-Gaussian state-space models. The scheme is a sparsity-promoting evolution of the likelihood consensus (LC) that uses the orthogonal matching pursuit (OMP), a B-spline dictionary, a distributed adaptive determination of the relevant state-space region, and an efficient binary representation of the LC expansion coefficients. Our simulation results show that a reduction of interagent communication by a factor of about 190 can be obtained without compromising the tracking performance.

***Index Terms*—** Target tracking, distributed particle filter, Bayesian filtering, likelihood consensus, sparsity.

1. INTRODUCTION

Distributed particle filters (DPFs) enable Bayesian target tracking in decentralized agent networks and possibly nonlinear and/or non-Gaussian state-space models [1–11]. A class of DPFs that is able to approximate the centralized particle filter with arbitrary accuracy uses the likelihood consensus (LC) scheme for a distributed calculation of the global likelihood function [5, 6]. The LC performs a dictionary expansion of the local log-likelihood function of each agent and disseminates and fuses the expansion coefficients by means of a consensus algorithm [6]. Generalizations of LC-based DPFs to scenarios with missed detections and clutter and/or a randomly appearing and disappearing target were presented in [12, 13].

Here, we propose an evolved LC methodology—dubbed "LC 2.0"—with significantly reduced communication cost. LC 2.0 features the following innovations: (i) A sparsity-promoting computation of the LC expansion coefficients via the orthogonal matching pursuit (OMP) [12, 14, 15]. Compared to the least-squares fit used so far, the OMP enables an easy specification and a reduction of the number of significant expansion coefficients, as we previously demonstrated in a different context [12]. (ii) Use of a B-spline dictionary [16, 17]. Contrary to the Fourier or monomial dictionary used previously [5, 6, 12, 13], the B-spline dictionary is localized, which is advantageous in view of the localized character of the posterior distribution. (iii) A restriction of the dictionary expansions to a time-dependent "region of interest," which is calculated in a distributed manner. (iv) Efficient binary representations of the expansion coefficients that have to be communicated. Our simulation results show that LC 2.0 can reduce interagent communication by a factor of about 190 without compromising the tracking performance.

This paper is organized as follows. Section 2 describes the system model and the local particle filters. Section 3 reviews the conventional LC scheme. The use of the OMP and of the B-spline dictionary is discussed in Sections 4 and 5, respectively. A distributed calculation of the region of interest is described in Section 6. Section 7 presents an efficient binary coefficient representation. Finally, simulation results are presented in Section 8.

This work was supported in part by the Austrian Science Fund (FWF) under grant P 32055-N31.

2. SYSTEM MODEL AND LOCAL PARTICLE FILTERS

We consider a target whose state $\mathbf{x}_n = (x_{n,1} \cdots x_{n,M})^{\mathrm{T}} \in \mathbb{R}^M$ evolves according to a state-transition probability density function (pdf) $f(\mathbf{x}_n|\mathbf{x}_{n-1})$, for $n \in \mathbb{N}$. The target state $\mathbf{x}_n$ is sensed by a decentralized network of S agents (sensors). Agent $s \in \{1, \ldots, S\}$ communicates with a certain set $\mathcal{N}_s \subseteq \{1, \ldots, S\} \setminus \{s\}$ of "neighboring" agents. The communication graph is assumed to be connected. At each time n, each agent s acquires a measurement $\mathbf{z}_n^{(s)}$ that is related to the target state $\mathbf{x}_n$ according to a *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 measurements of all agents at time n, $\mathbf{z}_n \triangleq (\mathbf{z}_n^{(1)\mathrm{T}} \cdots \mathbf{z}_n^{(S)\mathrm{T}})^{\mathrm{T}}$. Assuming that the agent measurements $\mathbf{z}_n^{(s)}$ are conditionally independent given $\mathbf{x}_n$, we have

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

At each time n, each agent s estimates the current target state $\mathbf{x}_n$ from the measurements of all agents 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, agent s runs a *local particle filter* in which the global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ underlying Bayesian estimation is represented by J pairs of particles and uniform weights, $\big\{\big(\mathbf{x}_n^{(s,j)}, w_n^{(s,j)} = 1/J\big)\big\}_{j=1}^{J}$. This particle representation is calculated time-recursively as follows [18]. 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 $f\big(\mathbf{x}_n|\mathbf{x}_{n-1}^{(s,j)}\big)$. 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, $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ is an approximation to the GLF $f(\mathbf{z}_n|\mathbf{x}_n)$ in (1) that involves the current measurements of all the agents, $\mathbf{z}_n$, and is calculated in a distributed way via the LC scheme reviewed in Section 3. 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 [18, 19]; this results in new particles $\mathbf{x}_n^{(s,j)}$ and weights $w_n^{(s,j)} = 1/J$. The overall recursion is initialized by particles $\mathbf{x}_0^{(s,j)}$ that are drawn from a prior pdf $f(\mathbf{x}_0)$, and by weights $w_0^{(s,j)} = 1/J$. Finally, agent s calculates a state estimate as the weighted sample mean of the predicted particles (before resampling), i.e, $\hat{\mathbf{x}}_n^{(s)} = \sum_{j=1}^{J} w_{n|n-1}^{(s,j)} \mathbf{x}_{n|n-1}^{(s,j)}$.

3. REVIEW OF THE LC SCHEME

Next, we briefly review the LC scheme [5, 6], which is used for a distributed calculation of the GLF approximation $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ in (2).

Consider

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

where (1) was used. Conversely, $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(SL_n(\mathbf{x}_n))$. Using a dictionary of "atoms" $\{\psi_k(\mathbf{x})\}_{k=1}^K$ that is identical for all agents, each agent s approximates its log-LLF by a linear combination of the atoms, i.e.,

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

Here, the local expansion coefficients $\big\{\alpha_n^{(s,k)}\big\}_{k=1}^K$ are calculated locally at each agent s using the local measurements $\mathbf{z}_n^{(s)}$, as described in Section 4. By inserting (4) into (3), we obtain

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

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

Approximations of the $\beta_n^{(k)}$ can be obtained at all agents in a distributed manner by means of K instances of the average consensus algorithm [20]. In consensus iteration $i \in \{1,2,\ldots\}$, agent s updates an iterated estimate of $\beta_n^{(k)}$ as

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

Here, the $\hat{\beta}_n^{(k,s')}[i-1]$, $s'\in\mathcal{N}_s$ were communicated to agent s by its neighbors $s'\in\mathcal{N}_s$, and the $\gamma_{s,s'}$ are suitably chosen weights, such as the Metropolis weights [20–22]. 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 substituted for the $\beta_n^{(k)}$ in (5) to obtain at each agent an approximation of $L_n(\mathbf{x}_n)$ and, in turn, an approximation $\hat{f}_s(\mathbf{z}_n|\mathbf{x}_n)$ of the GLF $f(\mathbf{z}_n|\mathbf{x}_n) = \exp(SL_n(\mathbf{x}_n))$. For $I\to\infty$, the consensus recursion would converge to $\beta_n^{(k)}$ because, as assumed in Section 2, the communication graph is connected [21].

In each iteration i, agent s has to broadcast the real numbers $\hat{\beta}_n^{(k,s)}[i]$, $k=1,\ldots,K$ to its neighbors $s'\in\mathcal{N}_s$. In the next four sections, we will propose modifications of the LC scheme that lead to a significant reduction of the communication cost. We note that while the total communication cost depends on the network topology, the relative reduction due to the proposed modifications does not. For reasons that will become clear in Section 6, we hereafter consider a time-dependent dictionary size K_n.

4. OMP-BASED LLF APPROXIMATION

In the original LC scheme [5, 6], the local expansion coefficients $\big\{\alpha_n^{(s,k)}\big\}_{k=1}^{K_n}$ involved in (4) are calculated at each agent s through a least squares (LS) fit of $\sum_{k=1}^{K_n}\alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$ to $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$, where both functions are evaluated (sampled) at the predicted particles, i.e., at $\mathbf{x}_n = \mathbf{x}_{n|n-1}^{(s,j)}$, $j=1,\ldots,J$. We now propose a sparsity-promoting alternative to LS that uses the OMP [14, 15]. Our goal is to obtain a low number of "significant" expansion coefficients and, thereby, a low communication cost. With a view toward the highly localized B-spline atoms proposed in Section 5, we sample $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ and $\sum_{k=1}^{K_n}\alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$ no longer at the predicted particles but *uniformly*. Indeed, we observed experimentally that this leads to significantly better performance when a B-spline dictionary is used. This is probably due to the fact that 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."

With uniform sampling, the evaluation points $\mathbf{x}_n^{(q)} = \big(x_{n,1}^{(q)}\cdots x_{n,M}^{(q)}\big)^{\mathrm{T}}$, $q = 1,\ldots,Q_n$ lie on a regular grid within an M-dimensional (M-D) *region of interest* (ROI)

$$\mathcal{R}_n \triangleq \big[a_n^{(1)},b_n^{(1)}\big]\times\ldots\times\big[a_n^{(M)},b_n^{(M)}\big] \subset \mathbb{R}^M. \quad (7)$$

(Recall that $\mathbf{x}_n\in\mathbb{R}^M$.) More specifically, in the mth coordinate direction, where $m\in\{1,\ldots,M\}$, we use $Q_n^{(m)}$ evaluation points uniformly spaced in the interval $\big[a_n^{(m)},b_n^{(m)}\big]$. The total number of M-D evaluation points $\mathbf{x}_n^{(q)}$ is thus $Q_n = \prod_{m=1}^{M}Q_n^{(m)}$. A distributed, particle-based method for calculating the ROI interval bounds $a_n^{(m)}$ and $b_n^{(m)}$ will be presented in Section 6.

The OMP is a greedy iterative algorithm that selects one atom per iteration; thus, the number of selected atoms equals the number of iterations. At agent s, in iteration $l\in\{1,2,\ldots\}$, the OMP selects the atom index k_l for which the discretized-atom vector $\boldsymbol{\psi}_{n,k} \triangleq \big(\psi_k(\mathbf{x}_n^{(1)})\cdots\psi_k(\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}}\in\mathbb{R}^{Q_n}$ best matches a residual vector $\boldsymbol{\rho}_{n,l-1}^{(s)}$ calculated at the previous iteration $l-1$, i.e.,

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

(Note that k_l depends on n and s, which is however not indicated for notational simplicity.) Then, the new residual $\boldsymbol{\rho}_{n,l}^{(s)}$ is formed as

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

where $\boldsymbol{\eta}_n^{(s)} \triangleq \big(\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(1)})\cdots\log f(\mathbf{z}_n^{(s)}|\mathbf{x}_n^{(Q_n)})\big)^{\mathrm{T}}\in\mathbb{R}^{Q_n}$ is a discretized-log-LLF vector, $\boldsymbol{\Psi}_{n,l}^{(s)}\in\mathbb{R}^{Q_n\times l}$ has columns $\boldsymbol{\psi}_{n,k_1}$, $\ldots$, $\boldsymbol{\psi}_{n,k_l}$ (this matrix depends on s because the OMP-based selection of its columns depends on the LLF at agent s), and $\boldsymbol{\Psi}_{n,l}^{(s)+} \triangleq \big(\boldsymbol{\Psi}_{n,l}^{(s)\mathrm{T}}\boldsymbol{\Psi}_{n,l}^{(s)}\big)^{-1}\boldsymbol{\Psi}_{n,l}^{(s)\mathrm{T}}$ is the Moore–Penrose pseudoinverse of $\boldsymbol{\Psi}_{n,l}^{(s)}$. We note that $\boldsymbol{\Psi}_{n,l}^{(s)}\mathbf{c}_{n,l}^{(s)} = \boldsymbol{\Psi}_{n,l}^{(s)}\boldsymbol{\Psi}_{n,l}^{(s)+}\boldsymbol{\eta}_n^{(s)}$ is the orthogonal projection of $\boldsymbol{\eta}_n^{(s)}$ onto the subspace of $\mathbb{R}^{Q_n}$ spanned by $\boldsymbol{\psi}_{n,k_1},\ldots,\boldsymbol{\psi}_{n,k_l}$. The residual is initialized as $\boldsymbol{\rho}_{n,0}^{(s)} = \boldsymbol{\eta}_n^{(s)}$.

Let $L_s\le K_n$ denote the number of OMP iterations performed at agent s, which is prespecified or defined by the condition that $\|\boldsymbol{\rho}_{n,l}^{(s)}\|$ falls below a positive threshold. Then the result of the OMP algorithm is the coefficient vector $\boldsymbol{\alpha}_n^{(s)} = \big(\alpha_n^{(s,1)}\cdots\alpha_n^{(s,K_n)}\big)^{\mathrm{T}}$ whose elements are $\alpha_n^{(s,k)} = \big(\mathbf{c}_{n,L_s}^{(s)}\big)_l$ for all $k=k_l$ with $l=1,\ldots,L_s$ and zero otherwise. Thus, the number of nonzero $\alpha_n^{(s,k)}$ equals L_s. The fact that a prescribed sparsity of $\boldsymbol{\alpha}_n^{(s)}$ is obtained by terminating the OMP after a fixed number of iterations is an advantage. Other sparse approximation methods such as ℓ_1-based methods [23] usually need careful parameter tuning to achieve a prescribed sparsity.

5. B-SPLINE DICTIONARY

We recall that agent s approximates the log-LLF $\log f\big(\mathbf{z}_n^{(s)}|\mathbf{x}_n\big)$ by the dictionary expansion $\sum_{k=1}^{K_n}\alpha_n^{(s,k)}\psi_k(\mathbf{x}_n)$ (see (4)) within the M-D ROI $\mathcal{R}_n$ defined in (7). In the original LC scheme, the dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^{K_n}$ consisted of monomial [5, 6] or Fourier [6, 12, 13] atoms. In this section, we propose the use of a dictionary of M-D B-splines with uniform knot spacing [16, 17]. The advantage of B-spline dictionaries is the localization of their atoms, which is desirable in view of the localization of the posterior pdf in the state space. We temporarily drop the time index n for notational simplicity.

Let us first consider 1-D B-spline dictionaries for the individual coordinate directions of the state space. A 1-D B-spline of degree $r \in \mathbb{N}_0$ is a piecewise polynomial function composed of polynomial segments of degree r. In particular, the cubic (i.e., $r=3$) 1-D B-spline prototype with knots positioned at the integers is an even function $\psi(x)$ with support $(-2,2)$ that is given by $\frac{2}{3}-|x|^2+\frac{1}{2}|x|^3$ for $0 \leq |x| < 1$, by $\frac{1}{6}(2-|x|)^3$ for $1 \leq |x| < 2$, and by zero otherwise. The atoms forming the 1-D B-spline dictionary for the mth coordinate direction, $\{\psi_{\tilde{k}}^{(m)}(x)\}_{\tilde{k}=1}^{\tilde{K}_m}$, are now defined by scaling and shifting the B-spline prototype $\psi(x)$ according to

$$\psi_{\tilde{k}}^{(m)}(x) = \psi\left(\frac{x-a^{(m)}-\tilde{k}\Delta d^{(m)}}{\Delta d^{(m)}}\right), \quad x \in \left[a^{(m)}, b^{(m)}\right], \quad (8)$$

for $\tilde{k}=1,\ldots,\tilde{K}_m$ and for each $m \in \{1,\ldots,M\}$. 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. Note that the $\psi_{\tilde{k}}^{(m)}(x)$ 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 ROI interval $\left[a^{(m)}, b^{(m)}\right]$ (see (7)) with spacing $\Delta d^{(m)}$.

The M-D B-spline atoms are then constructed as

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

with M-D index $\tilde{\mathbf{k}} \triangleq (\tilde{k}_1 \cdots \tilde{k}_M)^{\mathrm{T}}$ where $\tilde{k}_m \in \{1,\ldots,\tilde{K}_m\}$. The M-D B-spline dictionary $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ is finally 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 a regular M-D grid. We note that $a^{(m)}$, $b^{(m)}$, $\Delta d^{(m)}$, $\tilde{K}_m$, K, and $\{\psi_k(\mathbf{x})\}_{k=1}^{K}$ generally depend on the time index n, as discussed next.

6. DISTRIBUTED ROI ADAPTATION

The choice of the ROI $\mathcal{R}_n$ in (7) influences the accuracy and communication cost of the LC. We now present a distributed algorithm for calculating the ROI interval bounds $a_n^{(m)}$, $b_n^{(m)}$. The goal is to adaptively "zoom in" on the effective support of the current global posterior pdf $f(\mathbf{x}_n|\mathbf{z}_{1:n})$ in order to avoid using computation and communication resources to approximate the log-LLF on irrelevant parts of the state space. 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 $d_n^{(m)} \triangleq b_n^{(m)} - a_n^{(m)}$ at each time n in a distributed manner. This is done before the LC is performed, because the LC dictionary depends on the ROI. Note that $a_n^{(m)}$ 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 $b_n^{(m)} = \xi_n^{(m)} + d_n^{(m)}/2$.

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

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

where $x_{n|n-1,m}^{(s,j)}$ denotes the mth element of $\mathbf{x}_{n|n-1}^{(s,j)}$ and $\hat{x}_{n,m}^{(s)} \triangleq \frac{1}{J}\sum_{j=1}^{J} x_{n|n-1,m}^{(s,j)}$. Then, for each m, $\hat{x}_{n,m}^{(s)}$ and $\sigma_{n,m}^{(s)2}$ are averaged over all agents in a distributed manner. To this end, we perform $2M$ instances of the average consensus algorithm to compute approximations $\bar{\hat{x}}_{n,m}^{(s)}$ of $\bar{\hat{x}}_{n,m} \triangleq \frac{1}{S}\sum_{s=1}^{S}\hat{x}_{n,m}^{(s)}$ and $\overline{\sigma_{n,m}^{2}}^{(s)}$ of $\overline{\sigma_{n,m}^{2}} \triangleq \frac{1}{S}\sum_{s=1}^{S}\sigma_{n,m}^{(s)2}$, for $m=1,\ldots,M$. These approximations will be (slightly) different at different agents s. Because the LC requires the same ROI $\mathcal{R}_n$ at each agent, we next perform $2M$ instances of the maximum consensus algorithm [24] to calculate $\bar{\hat{x}}_{n,m}^{\max} \triangleq \max\{\bar{\hat{x}}_{n,m}^{(s)}\}_{s=1}^{S}$ 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 that is given by the diameter of the agent network [24]. The ROI center point $\boldsymbol{\xi}_n$ is now taken to be the vector with elements $\xi_n^{(m)} = \bar{\hat{x}}_{n,m}^{\max}$, $m=1,\ldots,M$. Furthermore, we choose each ROI extent $d_n^{(m)}$ as a function of $s_{n,m} \triangleq \sqrt{\overline{\sigma_{n,m}^{2}}^{\max}}$ in such a way that $\mathcal{R}_n$ tends to include all particles but is not unnecessarily large because this would result in an excessive dictionary size K_n. Thus, we set $d_n^{(m)} = \gamma\, s_{n,m}$, with a scaling factor $\gamma > 1$, if $\gamma\, s_{n,m} \geq d_{\min}^{(m)}$, and $d_n^{(m)} = d_{\min}^{(m)}$ otherwise. Here, thresholding $d_n^{(m)}$ at a lower bound $d_{\min}^{(m)}$ adds robustness in cases where the predicted particles $\{\mathbf{x}_{n|n-1}^{(s,j)}\}_{j=1}^{J}$ are highly concentrated in the state space.

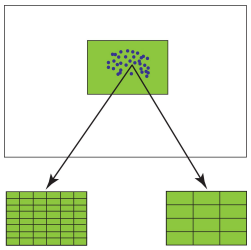

Fig. 1: An ROI $\mathcal{R}_n$ for dimension $M=2$ (green rectangle) is covered with B-spline atoms of two different densities. The small grid rectangles within the ROI indicate the effective supports of the B-spline atoms, and the blue dots indicate the predicted particles $\mathbf{x}_{n|n-1}^{(s,j)}$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

If the 1-D B-spline density $\kappa_{n,m}$, i.e., the number of 1-D B-spline atoms $\psi_{\tilde{k}}^{(m)}(x)$ per unit length in the mth coordinate direction, is specified, then the number $\tilde{K}_{n,m}$ of 1-D B-spline atoms follows as

$$\tilde{K}_{n,m} = \left\lceil \kappa_{n,m}\, d_n^{(m)} \right\rceil. \quad (9)$$

The overall dictionary size, i.e., the number of M-D B-spline atoms $\psi_k(\mathbf{x})$, $k=1,\ldots,K_n$, is then $K_n = \prod_{m=1}^{M}\tilde{K}_{n,m}$. Fig. 1 illustrates the covering of the ROI $\mathcal{R}_n$ with B-spline atoms using two different atom densities. A higher density enables a more accurate log-LLF approximation within $\mathcal{R}_n$ but implies a larger dictionary size K_n and, typically, a higher communication cost.

7. BINARY COEFFICIENT REPRESENTATION

In LC iteration i, each agent s broadcasts its iterated coefficient estimates $\hat{\beta}_n^{(k,s)}[i]$, $k=1,\ldots,K_n$ (see (6)) to the neighboring agents $s' \in \mathcal{N}_s$. If each $\hat{\beta}_n^{(k,s)}[i]$ is represented by a bit sequence of length n_{b}, then $K_n n_{\mathrm{b}}$ bits have to be broadcast by agent s in LC iteration i. 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. Let $L_s[i]$ denote the number of nonzero $\hat{\beta}_n^{(k,s)}[i]$. Before the first LC iteration $i=1$, the coefficient estimates are initialized as $\hat{\beta}_n^{(k,s)}[0] = \alpha_n^{(s,k)}$, and thus $L_s[0]$ equals the number of nonzero $\alpha_n^{(s,k)}$. In the course of the LC iterations $i=1,2,\ldots,I$, $L_s[i]$ will generally grow beyond $L_s[0]$ because at different agents s, the sets 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 $\hat{\beta}_n^{(k,s)}[i]$.

We now propose an efficient method for binary encoding of the indices k of the nonzero $\hat{\beta}_n^{(k,s)}[i]$. This method is specifically suited

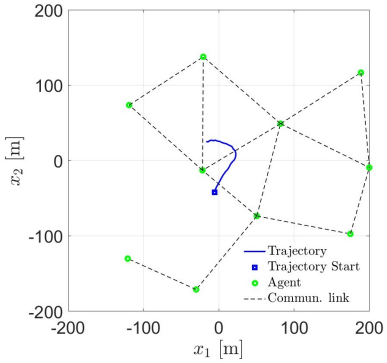

Fig. 2: Surveillance region, agent network, and target trajectory.

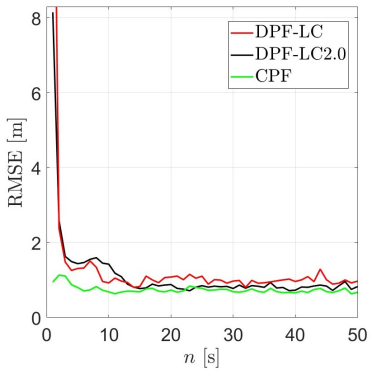

Fig. 3: Position RMSE versus time.

to the highly localized B-spline atoms. According to Section 5, 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 also be indexed by $\tilde{\mathbf{k}} = (\tilde{k}_1 \cdots \tilde{k}_M)^{\mathrm{T}}$ with $\tilde{k}_m \in \{1, \ldots, \tilde{K}_{n,m}\}$ (see (8)). The M-D indices $\tilde{\mathbf{k}}$ corresponding to the *nonzero* $\hat{\beta}_n^{(k,s)}[i]$ are then located in a—typically small—M-D "discrete hyperrectangle" $\mathcal{K}_n^{(s)}[i]$ that consists of all $\tilde{\mathbf{k}}$ with $\tilde{k}_m \in \{\tilde{l}_{n,m}^{(s)}[i], \ldots, \tilde{l}_{n,m}^{(s)}[i] + \Delta\tilde{k}_{n,m}^{(s)}[i]\}$ for $m = 1, \ldots, M$, where $\tilde{l}_{n,m}^{(s)}[i]$ 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.

Agent 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 a binary indicator vector of length $|\mathcal{K}_n^{(s)}[i]|$ whose kth bit is 1 if $\hat{\beta}_n^{(k,s)}[i] \neq 0$ and 0 otherwise. This requires $L_s[i]\, n_\mathrm{b} + |\mathcal{K}_n^{(s)}[i]|$ bits. In addition, agent s broadcasts the position and extent of $\mathcal{K}_n^{(s)}[i]$ via 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" $\boldsymbol{\Delta}_n^{(s)}[i] \triangleq \big(\Delta\tilde{k}_{n,1}^{(s)}[i] \cdots \Delta\tilde{k}_{n,M}^{(s)}[i]\big)^{\mathrm{T}}$. We need $\lceil \log_2(K_n) \rceil = \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m}) \big\rceil$ bits for $\tilde{\boldsymbol{l}}_n^{(s)}[i]$ (since $\tilde{\boldsymbol{l}}_n^{(s)}[i]$ may be one of $K_n = \prod_{m=1}^{M} \tilde{K}_{n,m}$ M-D indices $\tilde{\mathbf{k}}$) and $\lceil \log_2(N_\Delta) \rceil = \big\lceil \sum_{m=1}^{M} \log_2(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]) \big\rceil$ bits for $\boldsymbol{\Delta}_n^{(s)}[i]$ (since $\Delta\tilde{k}_{n,m}^{(s)}[i] \in \{1, \ldots, \tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i]\}$ and thus there are $N_\Delta \triangleq \prod_{m=1}^{M}(\tilde{K}_{n,m} - \tilde{l}_{n,m}^{(s)}[i])$ different $\boldsymbol{\Delta}_n^{(s)}[i]$). We conclude that the total number of bits broadcast by agent s in LC iteration i is

$$N^{(s)} = L_s[i]\, n_\mathrm{b} + |\mathcal{K}_n^{(s)}[i]| + \lceil \log_2(K_n) \rceil + \lceil \log_2(N_\Delta) \rceil .$$

8. SIMULATION RESULTS

We evaluate the performance and communication cost of LC 2.0 relative to the conventional LC. We simulated a target moving in the 2-D surveillance region $[-200\,\mathrm{m}, 200\,\mathrm{m}] \times [-200\,\mathrm{m}, 200\,\mathrm{m}]$. The target state $\mathbf{x}_n$ comprises position and velocity, and its evolution is modeled by the nearly constant velocity model [25, Ch. 6] with Gaussian driving noise (standard deviation $1/3$ m/s^2). The agent network consists of $S = 10$ agents. Fig. 2 shows the surveillance region, agent network, and target trajectory. Each agent s produces nonlinear range-bearing measurements $\mathbf{z}_n^{(s)}$ with Gaussian measurement noise (standard deviations $5/3$ m and $10/3^\circ$).

We compare a DPF using LC 2.0 (dubbed DPF-LC2.0), a DPF using the conventional LC (DPF-LC), and a centralized multisensor particle filter (CPF). All particle filters use $J = 10000$ particles. DPF-LC2.0 and DPF-LC perform $I = 20$ consensus iterations. DPF-LC2.0 uses B-spline atoms placed within the ROI, with $\tilde{K}_{n,1}$ and $\tilde{K}_{n,2}$ chosen adaptively according to (9) with $\kappa_{n,1} = \kappa_{n,2} = 1/20$. The ROI is calculated adaptively using $\gamma = 10$ and $d_{\min}^{(m)} = 20$. The log-LLF is sampled uniformly on the ROI with a density of one sample per meter. The number of OMP iterations is $L = \min\{5, K_n\}$, where $K_n = \tilde{K}_{n,1}\tilde{K}_{n,2}$. The binary wordlength for each nonzero coefficient is $n_\mathrm{b} = 32$. By contrast, DPF-LC employs LS-based calculation of the local expansion coefficients, using the ten dominant coefficients and setting the other coefficients to zero; $K = 1681$ Fourier atoms covering the entire surveillance region; and $n_\mathrm{b} = 32$ bits to represent each coefficient. We measure the tracking accuracy by the root mean square error (RMSE) of the position estimate and by the track loss percentage, based on 100 simulation runs performed over 50 time steps n. The RMSE is averaged over all agents and over all simulation runs for which it is smaller than 5 m for $n \geq 11$; the other simulation runs are considered track losses.

Fig. 3 shows that the RMSE of DPF-LC2.0 is similar to that of DPF-LC, and close to that of CPF. (We note that the parameters of the two DPFs were chosen to obtain similar RMSEs, as a basis for a meaningful comparison of the communication costs.) 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. The average communication cost per time step, agent, and LC iteration was measured as 284 bit for DPF-LC2.0 and 53792 bit for DPF-LC. Thus, the communication cost of DPF-LC2.0 is only about 0.53% of that of DPF-LC, corresponding to a reduction by a factor of about 190. We furthermore observed that, after a short initial phase, the number of B-spline atoms employed by DPF-LC2.0 was only $K_n = 4$ for almost all times and simulation runs.

9. CONCLUSION

The likelihood consensus (LC) scheme enables approximately Bayes-optimal distributed particle filtering in nonlinear and non-Gaussian agent networks. We improved the LC scheme by introducing the use of the OMP and a B-spline dictionary, a distributed adaptation of the region of interest, and efficient binary representations. In the resulting "LC 2.0" scheme, interagent communication is significantly reduced without a loss in tracking performance.

We remark that the proposed LC 2.0 scheme can also be used in other distributed filtering frameworks involving a factorizing global likelihood function, such as the distributed probabilistic data association (DPDA) filter [12] and the distributed Bernoulli filter [13]. An extended presentation, including the use of LC 2.0 for the DPDA filter and additional simulation results, is provided in [26].

10. REFERENCES

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

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

[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] 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.

[13] 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.

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

[15] 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.

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

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

[18] 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.

[19] 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.

[20] 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.

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

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

[23] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *PNAS*, vol. 100, no. 5, pp. 2197–2202, 2003.

[24] 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.

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

[26] E. Šauša, P. Rajmic, and F. Hlawatsch, "Distributed Bayesian target tracking with reduced communication: Likelihood consensus 2.0," *Signal Processing*, vol. 215, 109259, Feb. 2024.