Cooperative Simultaneous Localization and Tracking in Mobile Agent Networks

Florian Meyer, Ondrej Hlinka, Henk Wymeersch, Erwin Riegler, and Franz Hlawatsch

Abstract—We introduce a framework and methodology of cooperative simultaneous localization and tracking (CoSLAT) in decentralized mobile agent networks. CoSLAT provides a consistent combination of cooperative self-localization (CSL) and distributed target tracking (DTT). Multiple mobile targets and mobile agents are tracked using pairwise measurements between agents and targets and between agents. We propose a distributed CoSLAT algorithm that combines particle-based belief propagation with the likelihood consensus scheme and performs a bidirectional probabilistic information transfer between CSL and DTT. Simulation results demonstrate significant improvements in both self-localization and target tracking performance compared to separate CSL and DTT.

Index Terms—Agent networks, distributed target tracking, cooperative localization, belief propagation, message passing, factor graph, likelihood consensus, CoSLAT.

I. INTRODUCTION

A. Background and State of the Art

Cooperative self-localization (CSL) [1], [2] and distributed target tracking (DTT) [3] are key signal processing tasks in decentralized agent networks. Applications include surveillance [4], robotics [5], pollution source localization [6], environmental and agricultural monitoring [7], and chemical plume tracking [8]. In CSL, each cooperative agent (CA) measures quantities related to the location of neighboring CAs relative to the CA in question (e.g., involving distances or angles). By cooperating with other CAs, each CA is able to estimate its own location. In DTT, the CA measurements are related to the states of targets to be tracked. At each CA, estimates of the target states are cooperatively calculated from all CA measurements. CSL and DTT are related since, ideally, a CA needs to know its location to be able to contribute to DTT. This relation motivates the combined CSL-DTT method proposed in this paper, which achieves improved performance through a probabilistic information transfer between CSL and DTT.

For CSL of static CAs (hereafter termed “static CSL”), a factor graph framework and the nonparametric belief propagation (BP) algorithm are proposed in [8]. In [2], a distributed BP message passing algorithm for CSL of mobile CAs (hereafter termed “dynamic CSL”) is proposed. A message passing algorithm based on the mean field approximation is presented for static CSL in [9]. In [10], nonparametric BP is extended to dynamic CSL, and BP with a parametric message representation combined with censoring is considered. In [11], a particle-based BP method that uses a Gaussian approximation for the beliefs is proposed. The low-complexity method for dynamic CSL presented in [12] is based on the Bayesian filter and a linearized measurement equation.

For DTT, distributed particle filters (PFs) [13] are attractive since they are suited to nonlinear and non-Gaussian systems. Examples include [14] and [15]. In [14], consensus algorithms are used to calculate global weights—reflecting the measurements of all CAs—at each CA. In [15], the likelihood consensus (LC) scheme is proposed for a distributed approximate computation of the global (all-CAs) likelihood function, which is used at each CA for calculating global weights.

In the framework of simultaneous localization and tracking (SLAT), CAs track a target and localize themselves, using measurements of the distances between each CA and the target [16]. In contrast to dynamic CSL, the CAs are static and measurements of the distances between CAs are only used for initialization. A centralized particle-based SLAT method using BP is proposed in [17]. Distributed SLAT methods include a technique using a Bayesian filter and communication via a junction tree [18], iterative maximum likelihood methods [19], and variational filtering [20].

B. Contributions and Paper Organization

Extending our work in [21] and [22], we introduce the framework of cooperative simultaneous localization and tracking (CoSLAT). This framework provides a complete and consistent combination of dynamic CSL and DTT in decentralized mobile agent networks. In CoSLAT, single or multiple targets are tracked while simultaneously localizing the mobile CAs, using pairwise measurements between CAs and targets as well as between CAs. Thus, CoSLAT is different from SLAT in that it allows for CA mobility and uses pairwise measurements between the CAs also during runtime.

We first discuss two alternative particle implementations of BP for dynamic CSL [2], [10], one being a new implementation with reduced complexity. We then extend these implementations to CoSLAT. The resulting CoSLAT message passing algorithm is, to the best of our knowledge, the first method for simultaneous CSL and DTT in a fully dynamic setting. A key feature of CoSLAT is a bidirectional probabilistic information transfer between the CSL and DTT stages, which allows uncertainties in one stage to be taken into account by...
the other stage and thereby improves the performance of both stages. A difficulty is the fact that, due to the noncooperative nature of the targets, certain messages needed to calculate the target beliefs are not available at the CAs. The proposed CoSLAT algorithm employs the LC \cite{15} for a distributed calculation of these messages. The resulting combination of BP and LC may also be useful in other distributed inference problems involving noncooperative agents.

This paper is organized as follows. The CoSLAT system model is described in Section \textbf{II}. In Section \textbf{III} we present two implementations of BP message passing for dynamic CSL. Relevant background on DTT and LC is provided in Section \textbf{IV}. The CoSLAT message passing scheme and a corresponding distributed CoSLAT algorithm are presented in Section \textbf{V} and Section \textbf{VI} respectively. Some algorithmic variations and implementation aspects are discussed in Section \textbf{VII}. Finally, simulation results are presented in Section \textbf{VIII}.

\section{System Model}

We consider a decentralized network of mobile agents $k \in A$, as illustrated in Fig. 1. The set of all agents, $A \subseteq \mathbb{N}$, consists of the set of CAs, $S \subseteq A$, and the set of targets, $T = A \setminus S$. (We will use the indices $k \in A$, $l \in S$, and $m \in T$ to denote a generic agent, a CA, and a target, respectively.) The state of a target $k \in A$ at time $n \in \{0, 1, \ldots \}$, denoted $x_{k,n}$, consists of the current location and, possibly, motion parameters such as velocity \cite{23}. The states evolve according to

$$x_{k,n} = g(x_{k,n-1}, u_{k,n}), \quad k \in A, \quad (1)$$

where $u_{k,n}$ denotes driving noise with probability density function (pdf) $f(u_{k,n})$. The statistical relation between $x_{k,n-1}$ and $x_{k,n}$ defined by (1) can also be described by the state-transition pdf $f(x_{k,n}|x_{k,n-1})$.

The communication and measurement topologies of the network are described by sets $C_{l,n}$ and $M_{l,n}$ as follows. CA $l \in S$ is able to communicate with CA $l'$ if $l' \in C_{l,n} \subseteq S \setminus \{l\}$. Communication is always symmetric, i.e., $l \in C_{l,n}$ implies $l \in C_{l',n}$. Furthermore, CA $l \in S$ acquires a measurement $y_{l,k,n}$ relative to agent (CA or target) $k \in A$ if $k \in M_{l,n} \subseteq A \setminus \{l\}$. The targets are noncooperative in that they do not communicate with the CAs and do not acquire any measurements. We also define $M_{l,n}^S \triangleq M_{l,n} \cap S$ and $M_{l,n}^T \triangleq M_{l,n} \cap T$, i.e., the subsets of $M_{l,n}$ containing only CAs and only targets, respectively, and $S_{m,n} \triangleq \{ l \in S | m \in M_{l,n} \}$, i.e., the set of CAs that acquire measurements of target $m$. Note that $m \in M_{l,n}^T$ if and only if $l \in S_{m,n}$. We assume that $M_{l,n} \subseteq C_{l,n}$, i.e., if CA $l$ acquires a measurement relative to CA $l'$, it is able to communicate with CA $l'$. The sets $C_{l,n}$, $M_{l,n}$, $M_{l,n}^T$, $M_{l,n}^S$, and $S_{m,n}$ may be time-dependent. An example of communication and measurement topologies is given in Fig. 1.

We consider “pairwise” measurements $y_{l,k,n}$ that depend on the states $x_{l,n}$, $l \in S$ and $x_{k,n}$, $k \in M_{l,n}$, according to

$$y_{l,k,n} = h(x_{l,n}, x_{k,n}, v_{l,k,n}), \quad l \in S, \quad k \in M_{l,n}, \quad (2)$$

where $v_{l,k,n}$ is measurement noise with pdf $f(v_{l,k,n})$. An example is the scalar “noisy distance” measurement

$$y_{l,k,n} = \| \hat{x}_{l,n} - x_{k,n} \| + v_{l,k,n}, \quad (3)$$

where $\hat{x}_{k,n}$ represents the location of agent $k$ (this is part of the state $x_{k,n}$). The statistical dependence of $y_{l,k,n}$ on $x_{l,n}$ and $x_{k,n}$ is described by the local likelihood function $f(y_{l,k,n}|x_{l,n}, x_{k,n})$. We denote $x_n \triangleq \{ x_{k,n} \}_{k \in A}$ and $y_n \triangleq \{ y_{l,k,n} \}_{k \in A \setminus S, \ l \in M_{l,n}}$ the sets of all states and measurements, respectively at time $n$. Furthermore, $x_{l,n} \triangleq \{ x_{l,1}, \ldots, x_{l,n} \}$ and $y_{l,n} \triangleq \{ y_{l,1}, \ldots, y_{l,n} \}$.

We will make the following commonly used assumptions, which are reasonable in many practical scenarios \cite{3}.

(A1) All agent states are independent \textit{a priori} at time $n=0$, i.e., $f(x_0) = \prod_{k \in A} f(x_{k,0})$.

(A2) All agents move according to a memoryless walk, i.e.,

$$f(x_{l,n} | x_{l,n-1}) = f(x_{l,n} | x_{l,n-1}, x_{l,n-2}), \quad (A2)$$

(A3) The state transitions at the various agents are independent, i.e.,

$$f(x_{l,n}|x_{l,n-1}) = \prod_{k \in A} f(x_{k,n}|x_{k,n-1}). \quad (A3)$$

(A4) The current measurements $y_n$ are conditionally independent of all past measurements, $y_{l,n-1}$, and all past states, $x_{l,n-1}$, given the current states $x_n$, i.e.,

$$f(y_n | x_{l,n-1}, y_{l,n-1}) = f(y_n | x_n) \quad (A4)$$

(A5) The current states $x_n$ are conditionally independent of all past measurements, $y_{l,n-1}$, and all past states, $x_{l,n-1}$, given the current states $x_n$, i.e.,

$$f(x_n | x_{l,n-1}, y_{l,n-1}) = f(x_n | x_n) \quad (A5)$$

(A6) The measurements $y_{l,k,n}$ and $y_{l',k',n}$ are conditionally independent given $x_n$ unless $(l,k) = (l',k')$, and each measurement $y_{l,k,n}$ depends only on the states $x_{l,n}$ and $x_{k,n}$. Together with (A4), this leads to the following factorization of the “total” likelihood function:

$$f(y_{l,n} | x_{l,n}) = \prod_{n=1}^{t-1} \prod_{k \in S} \prod_{l' \in M_{l,n}} f(y_{l,k,n} | x_{l,n}, x_{l',n}). \quad (A6)$$

Finally, we assume that CA $l \in S$ knows the functional forms of its own state-transition pdf and initial state pdf as well as of the state-transition pdfs and initial state pdfs of all targets, i.e., $f(x_{k,n}|x_{k,n-1})$ and $f(x_{k,0})$ for $k \in \{l\} \cup T$.

\section{Dynamic CSL Using Belief Propagation}

In this section, as a basis for our CoSLAT algorithm, we first review BP for dynamic CSL \cite{23,10} and discuss two particle implementations, one being new.

In dynamic CSL, each CA $l \in S$ estimates its state $x_{l,n}$ from the past and present measurements of all CAs, $y_{l,n} \triangleq \{ y_{l',k,n} \}_{l' \in S, k' \in M_{l',n}, n \in \{1, \ldots, n\}}$. In particular, the minimum mean-square error (MMSE) estimator \cite{24} of $x_{l,n}$ is given by

$$\hat{x}_{l,n}^{\text{MMSE}} = \int x_{l,n} f(x_{l,n}|y_{l,n}^{S}) dx_{l,n}, \quad l \in S. \quad (4)$$

![Fig. 1. Agent network with CAs and targets.](image-url)
A. Dynamic CSL Message Passing Scheme

By Bayes’ rule and assumptions (A1)–(A6), the posterior pdf of the set of all CA states, \( P(S_1^n) \), is obtained up to an irrelevant factor as

\[
S_1^n = \prod_{l \in S} f(x_{l,0}) \prod_{n' = 1}^n \prod_{l, i' \in S} f(x_{l,n', n'}|x_{l,n-1'}) \times \prod_{l, i' \in M_{l,n'}} f(y_{l,i',n'}, x_{l,n'}),
\]

(5)

CAs with an informative prior pdf \( f(x_{l,0}) \) for which \( \prod_{n'} = 1 \) are referred to as anchors.

Two remarks are in order. First, for low complexity, communication requirements, and latency, messages are sent only during iterations. Second, as no information from the factor graph is involved in direct marginalization of \( f(S_1^n) \), \( l \in S \) can be obtained by executing a distributed iterative BP message passing scheme known as sum-product algorithm over a wireless network (SPAWN) [2]. At each time \( n, p \) message passing iterations are performed. The iterated belief of CA \( l \in S \) at time \( n \) and message passing iteration \( p \in \{1, \ldots, P\} \) is calculated as

\[
b^{(p)}(x_{l,n}) = \phi_{\rightarrow n}(x_{l,n}) \prod_{l', i' \in M_{l,n'}} \phi^{(p)}_{l' \rightarrow i'}(x_{l',n'}),
\]

(6)

where the “prediction message” \( \phi_{\rightarrow n}(x_{l,n}) \) is calculated from the state-transition pdf \( f(x_{l,n}|x_{l,n-1}) \) and the final belief at time \( n-1 \), \( b^{(p)}(x_{l,n-1}) \), as

\[
\phi_{\rightarrow n}(x_{l,n}) = \int f(x_{l,n}|x_{l,n-1}) b^{(p)}(x_{l,n-1}) dx_{l,n-1}
\]

(7)

and the “measurement messages” \( \phi_{l' \rightarrow i'}^{(p)}(x_{l',n'}) \) are calculated as

\[
\phi_{l' \rightarrow i'}^{(p)}(x_{l',n'}) = \int f(y_{l',i'|n'}, x_{l',n'}) b^{(p-1)}(x_{l,n'}) dx_{l,n'}.
\]

(8)

These messages and beliefs are depicted in Fig. 2.

Two remarks are in order. First, for low complexity, communication requirements, and latency, messages are sent only for the variable \( x_{l,n} \) to the factor \( f(x_{l,n}|x_{l,n-1}) \) equals the belief \( b^{(p)}(x_{l,n-1}) \), and \( \phi_{\rightarrow n}(x_{l,n}) \) in (7) (for \( n \) fixed) remains unchanged during all message passing iterations. Second, as no information from the factor graph \( f(y_{l,i'|n'}, x_{l',n'}) \) is used in the calculation of \( b^{(p-1)}(x_{l,n'}) \) according to (6) and (8), \( b^{(p-1)}(x_{l,n'}) \) in (8) is the extrinsic information with respect to \( f(y_{l,i'|n'}, x_{l',n'}) \).

By the SPAWN message passing scheme [2]–[6], an approximation \( b^{(p)}(x_{l,n}) \) of the marginal posterior \( f(x_{l,n}|S_1^n) \) at each CA \( l \in S \) is calculated in a distributed manner using only local communication with neighbors. Each CA \( l \) broadcasts its belief \( b^{(p)}(x_{l,n}) \) (calculated according to (6)) to all CAs \( l_i \) for which \( l \in M_{l_i,n} \) and uses the beliefs received from

![Fig. 2. CSL factor graph for CAs](image)

the neighboring CAs, \( b^{(p)}(x_{l,n}) \) for \( l' \in M_{l,n} \), and the local measurements \( y_{l,i'|n}, l' \in M_{l,n} \) to calculate the message passing iteration, \( b^{(p+1)}(x_{l,n}) \) for \( l \in M_{l,n} \), according to (8). The messages \( b^{(p+1)}(x_{l,n}) \) are then used by CA \( l \) to calculate the new \( b^{(p+1)}(x_{l,n}) \) according to (6), etc. However, direct calculation of (6)–(8) is still infeasible. Two feasible implementations will be presented in the next two subsections.

B. Nonparametric BP for Dynamic CSL

We first review nonparametric BP [3] for dynamic CSL [10]. Nonparametric BP uses a particle representation (PR) of beliefs and messages, and in a CSL scenario, provides fast convergence and high accuracy [2].

1) Message filtering: PR-based calculation of (7) and (8) amounts to the generic problem of obtaining a PR of \( \phi(x') = \int c(x', x) b(x) dx \) from a PR \( \{ (x^{(j)}, w^{(j)}) \}_{j=1}^J \) of \( b(x) \). Here, \( c(x', x) \) is (proportional to) a conditional pdf \( f(x'|x) \). If we are able to obtain a PR \( \{ (x^{(j)}, w^{(j)}, u^{(j)}) \}_{j=1}^J \) of \( c(x', x) \), then \( \{ (x^{(j)}, w^{(j)}) \}_{j=1}^J \) constitutes a PR of \( \phi(x') \). Typically, \( x' = r(x, q) \) with a known function \( r(\cdot, \cdot) \) and a random vector \( q \). Particles \( \{ (x^{(j)}, u^{(j)}, w^{(j)})\}_{j=1}^J \) corresponding to \( c(x', x) b(x) \) can then be obtained by first drawing particles \( \{ (x^{(j)}, q^{(j)})\}_{j=1}^J \) from \( f(x, q) \) and then calculating \( x^{(j)} = r(x^{(j)}, q^{(j)}) \), \( j = 1, \ldots, J \).
2) Message multiplication: A PR \( \{\{x_{l,m},u_{l,m}\}\}_{j=1}^{J} \) of \( b^{(p)}(x_{l,n}) \propto \phi_{n-\gamma}(x_{l,n}) \prod_{t' \in M_{l,n}} f^{(p)}(y_{l',n}|x_{l',n}) \) in \([5]\) can be obtained using importance sampling \([26]\). We first draw particles \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \) from the proposal distribution \( q(x_{l,n}) = \phi_{n-\gamma}(x_{l,n}) \). Corresponding weights \( w_{l,n}^{(j)} \) are then obtained by calculating \( w_{l,n}^{(j)} \propto b^{(p)}(x_{l,n}^{(j)})/q(x_{l,n}^{(j)}) \), i.e.,

\[
w_{l,n}^{(j)} = \prod_{t' \in M_{l,n}} f^{(p)}(y_{l',n}|x_{l',n}) ,
\]

and normalizing, i.e., \( \frac{w_{l,n}^{(j)}}{\sum_{j=1}^{J} w_{l,n}^{(j)}} \). Since only a PR of \( \phi_{n-\gamma}(x_{l,n}) \) is available, we substitute a kernel estimate \( \phi_{n-\gamma}(x_{l,n}) \) for \( \phi_{n-\gamma}(x_{l,n}) \) in \([9]\) and thus obtain \( \hat{w}_{l,n}^{(j)} = \prod_{t' \in M_{l,n}} f^{(p)}(y_{l',n}|x_{l',n}) \). This message multiplication operation is the most complex part of nonparametric BP; its complexity scales quadratically in the number of particles \( J \).

3) Estimation: Finally, from the PR \( \{\{x_{l,n}^{(j)},u_{l,n}^{(j)}\}\}_{j=1}^{J} \) of \( b^{(p)}(x_{l,n}) \), an approximation of the estimate \( x_{l,n}^{\text{MSE}} \) in \([4]\) is obtained as

\[
x_{l,n}^{\text{MSE}} \approx \sum_{j=1}^{J} w_{l,n}^{(j)} x_{l,n}^{(j)} ,
\]

(11)

C. A New Low-complexity Algorithm for Dynamic CSL

Next, using the “stacking approach” introduced in \([27]\), we develop a low-complexity alternative to nonparametric BP. Let \( X_{l,n}^{\text{MMSE}} \triangleq \{x_{l,n}^{(j)}\}_{j=1}^{J} \). Using \( \phi_{n-\gamma}(x_{l,n}) \) in \([6]\), one readily obtains

\[
b^{(p)}(x_{l,n}) \propto \prod_{t' \in \{l\}_{l}} f(y_{l',n}|x_{l',n}, x_{l,n}) b^{(p-1)}(x_{l',n}) ,
\]

(12)

and \( dx_{l,n}^{(p-1)} \propto \prod_{t' \in \{l\}_{l}} dx_{l',n}^{(p-1)} b^{(p-1)}(x_{l',n}) \). Based on \([12]\), we obtain a PR \( \{\{x_{l,n}^{(j)},u_{l,n}^{(j)}\}\}_{j=1}^{J} \) of \( b^{(p)}(x_{l,n}) \) from a PR \( \{\{x_{l,n}^{(j)},u_{l,n}^{(j)}\}\}_{j=1}^{J} \) of \( b^{(p)}(x_{l,n}) \) which is obtained via importance sampling using \( q(x_{l,n}) \) as proposal distribution. There is no need to draw particles \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \) from \( q(x_{l,n}) \) because such particles can be obtained simply by stacking the particles \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \) representing \( \phi_{n-\gamma}(x_{l,n}) \) (which were calculated as described in Section III-B) and the particles \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \) representing \( b^{(p-1)}(x_{l,n}) \), \( l' \in \{l\}_{l} \) (which were received from neighboring CAs). Using these particles \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \), we obtain weights \( w_{l,n}^{(j)} \) by calculating \( w_{l,n}^{(j)} \propto b^{(p)}(x_{l,n}^{(j)})/q(x_{l,n}^{(j)}) \), i.e.,

\[
\hat{w}_{l,n}^{(j)} = \prod_{t' \in \{l\}_{l}} f(y_{l',n}|x_{l',n}, x_{l,n}^{(j)}),
\]

and normalizing.

We note that a kernel estimate of a message \( \phi(x) \) is calculated from a PR \( \{\{x^{(j)},w^{(j)}\}\}_{j=1}^{J} \) of \( \phi(x) \) as

\[
\hat{\phi}(x) = \sum_{j=1}^{J} w^{(j)} K(x-x^{(j)}),
\]

(10)

with some kernel function \( K(x) \). The set \( \{\{x_{l,n}^{(j)},u_{l,n}^{(j)}\}\}_{j=1}^{J} \) is a PR of \( b^{(p)}(x_{l,n}) \). Hence, \( \{\{x_{l,n}^{(j)},u_{l,n}^{(j)}\}\}_{j=1}^{J} \) is a PR of \( b^{(p)}(x_{l,n}) \). This is because \( x_{l,n}^{(j)} \) can be obtained by discarding from \( x_{l,n}^{(j)} \) all vectors \( x_{l,n}^{(j)} \) with \( l' \neq l \), i.e., by discarding \( \{x_{l,n}^{(j)}\}_{j=1}^{J} \), which is the Monte Carlo implementation of the marginalization \( b^{(p)}(x_{l,n}) \). Finally, a resampling \([26]\) is performed to obtain equally weighted particles representing \( b^{(p)}(x_{l,n}) \). These particles are broadcast to all neighboring CAs \( l' \in \{l\}_{l} \), where they are used to calculate the belief \( b^{(p+1)}(x_{l,n}) \).

This algorithm avoids kernel estimation. Its complexity scales as \( O(J) \), i.e., only linearly in the number of particles \( J \). Note that the target distribution \( b^{(p)}(x_{l,n}) \) is of higher dimension than that of nonparametric BP, \( b^{(p)}(x) \) (cf. \([9]\)).

Nonetheless, we will see in Section VIII that using the same \( J \) as for nonparametric BP yields high accuracy.

IV. DTT USING LIKELIHOOD CONSENSUS

Another basis of our CoSLAT algorithm is DTT using the LC \([13]\), which will be reviewed next. In DTT, at time \( n \), the CAs \( l' \in S_{m,n} \) acquire measurements \( y_{l',m,n} \) associated with target \( m \in T \). Each CA \( l \in S \) then estimates all target states \( x_{m,n} \), \( m \in T \) from the measurements of all CAs \( l' \in S_{m,n} \) up to time \( n \), \( y_{l',m,n} \triangleq \{y_{l',m,n}'\}_{l' \in S_{m,n}, n' \in \{1,...,n\}} \). This is done, e.g., by means of the MMSE estimator

\[
x_{m,n}^{\text{MMSE}} \triangleq \int f(x_{m,n}|y_{m,1:n}; X_{m,n}^{S}) dx_{m,n} , \quad m \in T .
\]

(13)

This estimate also involves the set of CA states up to time \( n \), \( X_{m,n}^{S} = \{x_{m,n}'\}_{l \in S, n' \in \{1,...,n\}} \), which normally would have to be estimated separately using a CSL method. However, the DTT method reviewed in the following merely assumes that each CA \( l \in S \) knows its own state \( x_{l,n} \).

A. LC-based Distributed Particle Filter for DTT

The statistical relationship between the set of all measurements involving target \( m \), \( y_{m,1:n} = \{y_{l,m,n}'\}_{l \in S_{m,n}} \), and the target state \( x_{m,n} \) is described by the global likelihood function (GLF)

\[
G_{m,n}(x_{m,n}) \triangleq f(y_{m,n}|x_{m,n}; X_{m,n}^{S}) = \prod_{l \in S_{m,n}} f(y_{l,m,n}|x_{m,n}; x_{l,n}) ,
\]

(14)

where assumption (A6) was used in the last step. Note that here \( x_{l,n} \) is the known location of CA \( l \). Based on assumptions (A2)–(A5), the posterior pdf involved in \([13]\) can be calculated sequentially according to \([26]\)

\[
f(x_{m,n}|y_{m,1:n}; X_{m,n}^{S}) \propto G_{m,n}(x_{m,n}) \int f(x_{m,n-1} | y_{m,1:n-1}; X_{m,n-1}^{S}) dx_{m,n-1} .
\]

(15)

A feasible approximation of sequential state estimation as given by \([13]\) and \([15]\) is provided by the particle fil-
The weights $w_{1:n}^{(j)}$ are calculated by evaluating the GLF $G_{m,n}(x_{m,n})$ at the particles $x_{m,n}^{(j)}$ [25]. The functional form of $G_{m,n}(x_{m,n})$ can be provided to each CA by the LC scheme [15], [23]. This is based on the following finite-rank basis expansion approximation of each local log-likelihood function:

$$
\prod_{l \in S_{m,n}} f(y_{l,m,n}\mid x_{m,n}; x_l), \quad j = 1, \ldots, L.
$$

(19)

For later reference, we note that sequential Bayesian DTT according to [14] and [15] is equivalent to running BP on the factor graph shown in Fig. 3 [25]. Because of the tree structure of this graph, BP is performed noniteratively, i.e., the message passing procedure [44]–[46], is performed only once to calculate $b(x_{m,n})$. Furthermore, $b(x_{m,n})$ is exactly equal to $f(x_{m,n} \mid y_{m,1:n})$ and $G_{m,n}(x_{m,n})$ becomes available at each CA with only local communication.

### B. Message Passing Interpretation

For later reference, we note that sequential Bayesian DTT according to [14] and [15] is equivalent to running BP on the factor graph shown in Fig. 3 [25]. Because of the tree structure of this graph, BP is performed noniteratively, i.e., the message passing procedure [44]–[46], is performed only once to calculate $b(x_{m,n})$. Furthermore, $b(x_{m,n})$ is exactly equal to $f(x_{m,n} \mid y_{m,1:n})$ and $G_{m,n}(x_{m,n})$ becomes available at each CA with only local communication.

$$
\prod_{l \in S_{m,n}} f(y_{l,m,n}\mid x_{m,n}; x_l), \quad j = 1, \ldots, L.
$$

(19)

For later reference, we note that sequential Bayesian DTT according to [14] and [15] is equivalent to running BP on the factor graph shown in Fig. 3 [25]. Because of the tree structure of this graph, BP is performed noniteratively, i.e., the message passing procedure [44]–[46], is performed only once to calculate $b(x_{m,n})$. Furthermore, $b(x_{m,n})$ is exactly equal to $f(x_{m,n} \mid y_{m,1:n})$ and $G_{m,n}(x_{m,n})$ becomes available at each CA with only local communication.
The short notation $\phi \in S$ or $(b_n p) \in T$, the likelihood function $L, l \in S$ and $L \in S$ is omitted for simplicity. The upper three (black) dotted boxes correspond to the CSL part; the bottom (red) dotted box corresponds to the DTT part. Edges between black dotted boxes imply communication between CAs. Only messages and beliefs involved in the computation of $b^{(p)}(x_{1,n})$ and $b^{(p)}(x_{m,n})$ are shown. Edges with filled arrowheads depict particle-based messages and beliefs, while edges with filled arrowheads depict messages involved in the LC scheme.

to the DTT factor graph in Fig. 5 the likelihood function related to measurements of a target, $f(y_{l,m,n}|x_{m,n})$, is now a factor $f_{m,n}$ in Fig. 4 between a target state and a CA state. These factors enable a probabilistic information transfer between CSL-DTT and CSL-DTT (see Fig. 5), which is another reason to the CSL factor graph in Fig. 3, the likelihood function $L$ at each message passing iteration $p$.

and the measurement messages (cf. (8))

$$
\phi^{(p)}_{k \rightarrow m}(x_{m,n}) = \int f(y_{l,m,n}|x_{m,n}) \psi^{(p-1)}_{l \rightarrow m}(x_{l,n}) \, dx_{l,n}, \quad l = S_{m,n}, \quad m \in T.
$$

Here, $\psi^{(p-1)}_{l \rightarrow m}(x_{l,n})$ and $\psi^{(p-1)}_{l \rightarrow m}(x_{l,n})$ (constituting the extrinsic information) are given by

$$
\psi^{(p-1)}_{l \rightarrow m}(x_{l,n}) = \phi_{l \rightarrow m}(x_{m,n}) \prod_{k \in M_{l,m}, l \in T} \psi^{(p-1)}_{l \rightarrow m}(x_{l,n}).
$$

$$
\psi^{(p-1)}_{l \rightarrow m}(x_{l,n}) = \phi_{l \rightarrow m}(x_{m,n}) \prod_{l \in M_{l,m}, l \in T} \psi^{(p-1)}_{l \rightarrow m}(x_{l,n}).
$$

For reasons discussed in Section III-A the prediction messages in (23) and (7) and the CA-related measurement messages in (24) (cf. (8)) differ from the standard BP message passing rules, i.e., the extrinsic information is equal to the belief. The messages and beliefs involved in the calculation of $b^{(p)}(x_{1,n})$ and $b^{(p)}(x_{m,n})$ are depicted in Fig. 4.

In the "pure DTT" case considered in Section VI CA $l \in S_{m,n}$ knows its own true state, $x_{m,n}^{\text{true}}$, and thus $\psi^{(p-1)}_{m \rightarrow l}(x_{l,n})$ is replaced by $\delta(x_{l,n} - x_{m,n}^{\text{true}})$. Hence, (25) yields $\phi^{(p)}_{m \rightarrow l}(x_{m,n}) = f(y_{l,m,n}|x_{m,n};x_{m,n}^{\text{true}})$, as was claimed in Section IV.B.

VI. DISTRIBUTED COSLAT ALGORITHM

We will next devise a distributed CoSLAT algorithm that combines particle-based BP—i.e., a particle-based implementation of (21) —with the LC. This algorithm requires only local communication between neighboring CAs. The distributed calculation of the target beliefs, CA beliefs, and extrinsic information will be discussed in separate subsections.
A. Distributed Calculation of the Target Beliefs

Estimation of the target states $x_{m,n}$, $m \in \mathcal{T}$ from $y_{1:n}$ according to (20) essentially amounts to a computation of $f(x_{m,n}|y_{1:n})$. The following discussion describes the calculations associated with the red dotted box in Fig. 4.

The target belief $b^{(p)}(x_{m,n}, p \in \{1, \ldots, P\}$ approximating $f(x_{m,n}|y_{1:n})$ is given, up to a factor, by (see (22))

$$b^{(p)}(x_{m,n}) \propto \phi_{m,n}(x_{m,n}) \Phi^{(p)}(x_{m,n}),$$

(28)

with (recalling (23))

$$\phi_{m,n}(x_{m,n}) = \int f(x_{m,n}|x_{m,n-1}) b^{(P)}(x_{m,n-1}) dx_{m,n-1}$$

(29)

and

$$\Phi^{(p)}(x_{m,n}) \triangleq \prod_{l \in S_{m,n}} \phi^{(p)}_{l,m}(x_{m,n}).$$

(30)

The key observation now is that expression (28) along with (29) is of the same form as the DTT recursion (15), but with the GLF $G_{m,n}(x_{m,n})$ replaced by the message product $\Phi^{(p)}(x_{m,n})$. The belief $b^{(p)}(x_{m,n-1})$ occurring in (29) was calculated by each CA at time $n-1$; using this belief, the CA is able to calculate the message $\phi_{m,n}(x_{m,n})$ involved in (28).

Regarding the message product $\Phi^{(p)}(x_{m,n})$, the individual messages $\phi^{(p)}_{l,m}(x_{m,n})$ (cf. (30)) involve the extrinsic informations $\psi_{l,m}(x_{m,n})$ (see (23)); calculation of the latter will be discussed in Section VII-C. However, because the targets do not cooperate, at each CA at most one message $\phi_{l,m}(x_{m,n})$ is available for (given) feedback. Thus, the overall message product $\Phi^{(p)}(x_{m,n})$ is not available at the CAs.

1) Distributed Calculation of $\Phi^{(p)}(x_{m,n})$: To solve this problem, we embed a variant of the LC in a particle implementation of BP. In analogy to (16), we approximate log $\Phi^{(p)}_{l,m}(x_{m,n})$ by a finite-rank basis expansion:

$$\log \Phi^{(p)}_{l,m}(x_{m,n}) \approx \sum_{r=1}^{R} \beta^{(p)}_{l,m,r}(y_{1:m,n}) \varphi_r(x_{m,n}),$$

(31)

where $\varphi_r$, $r \in \{1, \ldots, R\}$ are basis functions, and $\beta^{(p)}_{l,m,r}(y_{1:m,n})$ is the rth instance at CA $l \in S_{m,n}$ of an average consensus or gossip algorithm (30), (31). These algorithms are iterative; the rth instance at CA $l \in S_{m,n}$ is initialized with $\beta^{(p)}_{l,m,r}(y_{1:m,n})$ and converges to the average $\sum_{l \in S_{m,n}} \beta^{(p)}_{l,m,r}(y_{1:m,n})/|S| = \beta^{(p)}_{l,m,p}(y_{1:m,n})$ if the network’s communication graph is connected. Only local communication is required: at each iteration, $R$ real values are broadcast by each CA to neighboring CAs (30), (31). After convergence, assuming that the number of CAs $|S|$ is known at each CA, the $B^{(p)}_{l,m,p}$ and hence, due to (32), an approximation $\tilde{\Phi}^{(p)}_{m,n}(x_{m,n})$ of the functional form of $\Phi^{(p)}(x_{m,n})$ is available at each CA.

Note that $|S|$ can be determined in a distributed way by using another consensus or gossip algorithm. The overall algorithm is summarized in what follows.

**Algorithm 1: Distributed Computation of $\Phi^{(p)}_{m,n}(\cdot)$**

At time $n$ and message passing iteration $p$, CA $l$ performs the following operations:

**Step 1:** For $l \in S_{m,n}$, $\{\beta^{(p)}_{l,m,r}(y_{1:m,n})\}_{r=1}^{R}$ in (31) is calculated by LS fitting of $\sum_{r=1}^{R} \beta^{(p)}_{l,m,r}(y_{1:m,n}) \varphi_r(x_{m,n})$ to log $\phi_{l,m}(x_{m,n})$, using a closed-form expression of $\phi_{l,m}(x_{m,n})$, and as reference points, particles $\tilde{x}_{m,n} \in S_{m,n}$ representing $\phi_{l,m}(x_{m,n})$. For $l \notin S_{m,n}$, $\beta^{(p)}_{l,m,r}(y_{1:m,n}) = 0$ for all $r \in \{1, \ldots, R\}$.

**Step 2:** For each $r \in \{1, \ldots, R\}$, $B^{(p)}_{l,m,r}$ in (34) is calculated by means of a consensus or gossip algorithm. This step requires communication between neighboring CAs.

**Step 3:** The desired approximation of $\Phi^{(p)}_{m,n}(\cdot)$ is obtained as $\tilde{\Phi}^{(p)}_{m,n}(\cdot) = \exp \left( \sum_{r=1}^{R} B^{(p)}_{l,m,r} \varphi_r(x_{m,n}) \right)$.

The most complex part of the LC algorithm is the LS fitting in Step 1. If a QR decomposition is used, the complexity of LS fitting grows linearly with the number of particles $J$ and cubically with the number of basis functions $R$. (34)

2) Distributed Calculation of $b^{(p)}(x_{m,n})$: After execution of Algorithm 1, $\tilde{\Phi}^{(p)}_{m,n}(\cdot)$ for each target $m \in \mathcal{T}$ is available at each CA $l \in S$. Based on (28), each CA is now able to obtain a PR of the target belief $b^{(p)}(x_{m,n})$. This is done via importance sampling with proposal distribution $\phi_{l,m}(x_{m,n})$. First, based on (29), particles $\{x^{(j)}_{m,n}, j = 1, \ldots, J\}$ representing $\phi_{l,m}(x_{m,n})$ are calculated from particles representing $b^{(p)}(x_{m,n-1})$ by means of message filtering (cf. Section III-B3). Next, weights $\tilde{w}^{(j)}$ in (31) are calculated as $\tilde{w}^{(j)} \triangleq \Phi^{(p)}_{m,n}(x^{(j)}_{m,n})$ followed by normalization. Finally, resampling is performed to obtain equally weighted particles representing $b^{(p)}(x_{m,n})$.

3) Probabilistic Information Transfer: According to (25), the messages $\phi^{(p)}_{l,m}(x_{m,n})$ occurring in $\Phi^{(p)}_{m,n}(x_{m,n})$ involve the extrinsic informations $\psi_{l,m}(x_{l:m,n})$ of all CAs $l$ observing target $m$, i.e., $l \in S_{m,n}$. Therefore, they constitute an information transfer from the CSL part of CoSLAT to the DTT part (cf. the directed edges
entering the red dotted box in Fig. 4). The estimation of target state $x_{m,n}$ is based on the belief $b^{(p)}(x_{m,n})$ as given by (28), and thus on $\Phi^{(p)}_{m,n}(x_{m,n})$. This improves on pure DTT because probabilistic information about the states of the CAs $l \in S_m$, provided by $\psi_{l-m}(x_{m,n})$ —is taken into account. By contrast, pure DTT uses the GLF $G_{m,n}(x_{m,n})$ instead of $\Phi^{(p)}_{m,n}(x_{m,n})$ (see [15]). According to [15], this presupposes that the CA states are known. In separate CSL–DTT, estimates of the CA states provided by CSL are used, rather than probabilistic information about the CA states as is done in CoSLAT. The improved accuracy of target state estimation achieved by our CoSLAT algorithm compared to separate CSL–DTT will be demonstrated in Section VII-A.

### B. Distributed Calculation of the CA Beliefs

For distributed calculation of the CA belief $b^{(p)}(x_{l,n})$, $l \in S$, the following information is available at CA $l$: (i) equally weighted particles representing $\psi_{l-m}(x_{m,n})$ for all targets $m \in T$ (which were calculated as described in Section VI-A and VI-C); (ii) equally weighted particles representing $b^{(p-1)}(x_{l,n})$ for all neighboring CAs $l' \in S_{l}$ (which were received from these CAs); and (iii) a PR of $b^{(p)}(x_{l-n})$ (which was calculated at time $n-1$). Using this information and the measurements $y_{l,k,m}$, $k \in M_{l}$, a PR $\{x_{l,n}^{j,p}(\omega_{l,n})\}_{j=1}^{J}$ of $b^{(p)}(x_{l,n})$ can be calculated in a distributed manner by employing (21), using nonparametric BP for mobile CAs as reviewed in Section III-B or the new low-complexity method presented in Section III-C. Finally, resampling is performed to obtain equally weighted particles representing $b^{(p)}(x_{l,n})$. This calculation of the CA beliefs improves on pure CSL as reviewed in Section III in that it uses the probabilistic information about the states of the targets $m \in T$ provided by the messages $\psi_{m,n}(x_{m,n})$ (cf. (22)). The improved accuracy of CA state estimation will be demonstrated in Section VIII.

### C. Distributed Calculation of the Extrinsic Informations

Since (26) is analogous to (21) and (27) to (22), particles for $\psi_{l-m}(x_{m,n})$ or $\psi_{l-n}(x_{m,n})$ can be calculated similarly as for the corresponding belief. However, the following shortcut reuses previous results. To obtain particles for $\psi_{m,n}(x_{m,n})$, we proceed as for $b^{(p)}(x_{m,n})$ (see Section VI-A2) but replace $\psi^{(p)}_{m,n}(x_{m,n})$ with $\exp(\sum_{j=1}^{J}B_{m,n}(x_{m,n}))$, which is based on the relation $\psi_{m,n}(x_{m,n}) \propto \Phi^{(p)}_{m,n}(x_{m,n})/\phi^{(p)}_{m,n}(x_{m,n})$, cf. (22) and (27). Here, $B_{m,n}$ and $\phi_{m,n}(x_{m,n})$ are already available locally from the calculation of $b^{(p)}(x_{m,n})$.

### D. Statement of the CoSLAT Algorithm

The proposed distributed CoSLAT algorithm is obtained by combining the operations discussed in Sections VI-A through VI-C as summarized in the following.

**ALGORITHM 2: DISTRIBUTED COSLAT ALGORITHM**

**Initialization:** The recursive algorithm described below is initialized at time $n = 0$ with particles $\{x_{l,n}^{(i)}\}_{j=1}^{J}$ drawn from a prior pdf $f(x_{l,n})$, for $k \in \{l\} \cup T$.

**Recursion at time $n$:** At CA $l$, equally weighted particles $\{\tilde{x}_{l,n}^{j}\}_{j=1}^{J}$ representing the beliefs $b^{(p)}(x_{l,n-1})$ with $k \in \{l\} \cup T$ are available (these were calculated at time $n-1$). At time $n$, CA $l$ performs the following operations.

1. **Prediction:** From $\tilde{x}_{l,n}^{j}$, PRs $\{x_{l,n}^{j}\}_{j=1}^{J}$ of the prediction messages $\phi_{l,n}(x_{l,n})$, $k \in \{l\} \cup T$ are calculated via message filtering (see Section III-B1), based on the state-transition model in [1]. That is, $\tilde{x}_{l,n}^{j} = g(\tilde{x}_{l,n-1}^{j}, u_{k,n}^{j})$, where the particles $\{u_{k,n}^{j}\}_{j=1}^{J}$ are drawn from $f(u_{l,n})$.

2. **BP message passing:** For each $k \in \{l\} \cup T$, the belief is initialized as $b^{(0)}(x_{l,n}) = \phi_{l,n}(x_{l,n})$, in the sense that the PR of $\phi_{l,n}(x_{l,n})$ is used as PR of $b^{(0)}(x_{l,n})$. Then, for $p = 1, \ldots, P$:

   a) For each target $m \in T$, $\tilde{\phi}_{l,n}^{p}(\cdot)$ is computed by executing Algorithm 1. Next, a PR $\{x_{l,n}^{p}(m), w_{l,n}^{p}(m)\}_{j=1}^{J}$ of $b^{(p)}(x_{l,n})$ in (22) is obtained via importance sampling with proposal distribution $\phi_{l,n}(x_{l,n})$ (see Section VI-A2). That is, using the targets $\{x_{l,n}^{p}(m), w_{l,n}^{p}(m)\}_{j=1}^{J}$ of $b^{(p)}(x_{l,n})$ (calculated in Step 1), weights $\{w_{l,n}^{(p)}(m)\}_{j=1}^{J}$ are obtained by calculating $\tilde{\phi}_{l,n}^{p}(\cdot)$ and normalizing.

   b) For each target $m \in M_{T}$, a PR of $\psi_{l,m}^{p}(x_{l,m})$ is calculated in a similar manner (see Section VI-C).

   c) A PR $\{x_{l,n}^{j}(m), w_{l,n}^{j}(m)\}_{j=1}^{J}$ of $b^{(p)}(x_{l,n})$ is calculated by implementing (21) as described in Section VI-C. This involves equally weighted particles of all $b^{(p-1)}(x_{l,n})$, $k \in M_{l}$.

   d) For each $m \in M_{T}$, a PR of $\psi_{l,m}^{p}(x_{l,m})$ is calculated in a similar manner.

   e) For all PRs calculated in Steps 2a–2d, resampling is performed to obtain equally weighted particles.

   f) The equally weighted particles of $b^{(p)}(x_{l,n})$ calculated in Step 2e are broadcast to all CAs $l' \in M_{l}$, and equally weighted particles of $b^{(p)}(x_{l,n})$ are received from the neighbors. Thus, at this point, equally weighted particles $\{x_{l,n}^{j}(m)\}_{j=1}^{J}$ of the following quantities are available at CA $l$: $b^{(p)}(x_{l,n})$, for $k \in \{l\} \cup T \cup M_{T}$, $\psi_{l,m}^{p}(x_{l,m})$ for $m \in M_{T}$, and $\psi_{l,m}^{p}(x_{l,m})$ for $m \in M_{l}$.

3. **Estimation:** For $k \in \{l\} \cup T$, an approximation of the local MMSE state estimate $\hat{x}_{l,n}^{K_{m}}$ in (20) is computed from the PR $\{x_{l,n}^{j}(m)\}_{j=1}^{J}$ of $b^{(p)}(x_{l,n})$ according to

$$\hat{x}_{l,n}^{K_{m}} = \sum_{j=1}^{J} w_{l,n}^{j}(m) x_{l,n}^{j}(m), \quad k \in \{l\} \cup T.$$
then many initialization values \( \beta_{l,m,n}^{(p,r)}(y_{l,m,n}) \) are zero. We therefore introduce a modification, termed local distributed tracking (LDT), in which \( b^{(p)}(x_{m,n}) \) for a target \( m \in T \) is calculated via (22) only at CAs \( l \) that acquire a measurement of the target, i.e., \( l \in S_{m,n} \). This corresponds to calculating \( \mathbf{33} \) instead of \( \mathbf{43} \). The convergence is here faster due to the smaller “consensus network” \( l \in S_{m,n} \) instead of \( l \in S \) and the fact that zero initialization values are avoided. LDT presupposes that the communication graph of the network formed by all CAs \( l \in S_{m,n} \) is connected. To ensure that CAs \( l' \in S_{m,n+1} \setminus S_{m,n} \), i.e., with \( l' \notin S_{m,n} \) but \( l' \in S_{m,n+1} \) obtain the information needed to track target \( m \) at time \( n+1 \), each CA \( l \in S_{m,n} \) broadcasts \( b^{(p)}(x_{m,n}) \) (calculated as described in Section VI.A) to its neighbors \( l' \in C_{l,n} \). Using \( b^{(p)}(x_{m,n}) \), neighboring CAs \( l' \in S_{m,n+1} \setminus S_{m,n} \) are then able to calculate \( \beta_{l',m,n}^{(p)}(x_{l',m,n}) \) (see \( \mathbf{7} \) and Section III-B1) and to track target \( m \) at time \( n+1 \) according to (22).

LDT has certain drawbacks. First, only CAs \( l \in S_{m,n} \) obtain an estimate of the state of target \( m \). (Equivalently, each CA \( l \in S \) tracks only targets \( m \in M_{l,n}^{(T)} \).) Second, the size of the consensus network, \( |S_{m,n}| \), has to be estimated at each time \( n \). Third, in agent networks with few communication links, it is possible that a CA \( l' \in S_{m,n+1} \setminus S_{m,n} \) cannot communicate with any CA \( l \in S_{m,n} \) at time \( n \), i.e., \( l \notin C_{l,n} \). Then, CA \( l' \) does not obtain \( b^{(p)}(x_{m,n}) \) and cannot track target \( m \) at time \( n+1 \), even though it acquired a corresponding measurement at time \( n \). However, in many scenarios, the communication regions of the CAs are significantly larger than their measurement regions. The situation described above is then very unlikely.

B. Alternative Processing at \( n = 1 \)

In the CoSLAT algorithm, a PR of \( b^{(p)}(x_{k,n}) \) is calculated via importance sampling with \( \phi_{m,n}(x_{k,n}) \) used as proposal distribution \( q(x_{k,n}) \). At time \( n = 1 \), \( \phi_{m,n}(x_{k,n}) = \int f(x_{k,n}|x_{0}) \cdot f(x_{0}) \, dx_{0} \). Often, insufficient prior information about the substate \( x_{k,n} \) actually involved in the measurement \( \mathbf{2} \) is available, and thus a (partly) uninformative prior pdf \( f(x_{k,n}) \) is used. (An example is given by agents with an uninformative location prior and measurements \( y_{l,k,n} \) that depend only on the locations of agent \( k \in A \) and CA \( l \in S \), such as, e.g., in \( \mathbf{3} \).) This implies that \( \phi_{m,n}(x_{k,n}) \) is (partly) uninformative as well, and thus widely spread particles are generated. This is especially problematic for the LC-based calculation of the target beliefs, since the particles of the proposal distribution are used as reference points for LS fitting (see Section VI.A1). Here, uninformative reference points tend to lead to a poor approximation of \( \Phi_{m,n}(x_{u,n}) \).

We therefore propose an alternative processing at time \( n = 1 \) that works with a moderate number of particles and can be used for DTT, dynamic CSL, and CoSLAT. We adopt the model \( \phi_{m,n}(x_{k,1}) = f(x_{k,1}|x_{k,0}) \), where \( f(x_{k,1}) \) is an uninformative pdf of the location \( x_{k,1} \) (e.g., uniform on the entire localization region) and \( f(x_{k,1}) \) is an informative pdf of the complementary subvector \( x_{k,1} \) of \( x_{k,1} \) (e.g., Gaussian with small variance). (In CSL and CoSLAT, this alternative processing is employed only by CAs with an uninformative prior. Anchors use the standard proposal distribution \( \phi_{m,n}(x_{l,0}) = \int f(x_{l,1}|x_{l,0}) f(x_{l,0}) \, dx_{l,0} \).) Now, instead of using \( \phi_{m,n}(x_{k,1}) = f(x_{k,1})f(x_{k,1}|x_{k,1}) \) as proposal distribution, we use

\[
q^{(p)}(x_{k,1}) = \phi_{k' \rightarrow k}^{(p)}(x_{k,1}) f(x_{k,1}),
\]  

(35)

with some \( k' \in B_{k,n} \), where \( B_{k,n} \subset M_{k,n} \) if \( k \in S \) and \( B_{k,n} = S_{k,n} \) if \( k \in T \) (the choice of \( k' \) will be discussed later). Here, \( \phi_{k' \rightarrow k}(x_{k,1}) \) is a function of \( x_{k,1} \) but not of \( x_{k,1} \); thus, it is always true if the measurements \( y_{l,m,n} \) depend only on the locations \( x_{l,n} \) and \( x_{k,n} \). Thus, hereafter we will write \( \phi_{k' \rightarrow k}(x_{k,1}) \) instead of \( \phi_{k' \rightarrow k}(x_{k,1}) \). Replacing \( f(x_{k,1}) \) by \( \phi_{k' \rightarrow k}(x_{k,1}) \) is reasonable because \( \phi_{k' \rightarrow k}(x_{k,1}) \) is more informative than \( f(x_{k,1}) \). Particles \( \{x_{k,1}\}_{j=1}^{l} \) drawn from \( q^{(p)}(x_{k,1}) \) can be obtained by stacking particles \( \{x_{k,1}\}_{j=1}^{l} \) drawn from \( \phi_{k' \rightarrow k}(x_{k,1}) \) and particles \( \{x_{k,1}\}_{j=1}^{l} \) drawn from \( f(x_{k,1}) \). Typically, drawing particles from \( f(x_{k,1}) \) is trivial. Equally weighted particles representing \( \phi_{k' \rightarrow k}(x_{k,1}) \) can be calculated from equally weighted particles representing \( \phi_{k' \rightarrow k}(x_{k,1}) \) by using message filtering as described in Section III-B1.

A PR \( \{x_{k,1}, w_{k,1}\}_{j=1}^{l} \) of \( (x_{k,1}) \), \( k \in A \) is obtained by means of importance sampling using the proposal distribution \( q^{(p)}(x_{k,1}) \) in (35): particles \( \{x_{k,1}\}_{j=1}^{l} \) are drawn from \( q^{(p)}(x_{k,1}) \), and weights \( \{w_{k,1}\}_{j=1}^{l} \) are calculated according to (cf. \( \mathbf{21} \) and \( \mathbf{22} \))

\[
\frac{\phi_{m,n}(x_{k,1})}{\phi_{k' \rightarrow k}^{(p)}(x_{k,1})} = \frac{f(x_{k,1}) \cdot \phi_{k' \rightarrow k}(x_{k,1})}{\phi_{k' \rightarrow k}^{(p)}(x_{k,1})} = \frac{f(x_{k,1})}{\phi_{k' \rightarrow k}^{(p)}(x_{k,1})} \prod_{k' \in B_{k,n}} \phi_{k' \rightarrow k}^{(p)}(x_{k,1}),
\]

followed by normalization.

To make \( q^{(p)}(x_{k,1}) \) in (35) maximally informative, we choose \( \hat{k}' = \arg \min_{k' \in B_{k,n}} \sigma_{k' \rightarrow k}^{2} \).

(36)

Here, \( \sigma_{k' \rightarrow k}^{2} \) is the empirical variance of \( \phi_{k' \rightarrow k}(x_{k,1}) \), which is calculated from equally weighted particles \( \{x_{k,1}\}_{j=1}^{l} \) representing \( \phi_{k' \rightarrow k}(x_{k,1}) \) as

\[
\sigma_{k' \rightarrow k}^{2} = \frac{1}{l} \sum_{j=1}^{l} \left\| x_{k,1}^{(j)} - \mu_{k' \rightarrow k} \right\|^{2},
\]

with \( \mu_{k' \rightarrow k} = \frac{1}{l} \sum_{j=1}^{l} x_{k,1}^{(j)} \).

For \( k = m \in T \), \( S_{k,n} \) becomes \( \hat{l} = \arg \min_{l \in S_{m,n}} \sigma_{k' \rightarrow m}^{2} \). This has to be calculated in a distributed manner, and the particles corresponding to the optimum \( \hat{l} \) are of \( x_{k,1} \) have to be provided to all \( l' \in S \). Both tasks can be done by using the following algorithm based on the min-consensus (35).

**Algorithm 3: Distributed Calculation of \( \Phi_{m,n}(x_{u,n}) \)**

At time \( n = 1 \) and message passing iteration \( p \), CA \( l \) performs the following operations:

**Step 1:** For each \( m \in T \), equally weighted particles \( \{x_{k,1}\}_{j=1}^{l} \) representing the (locally available) message \( \phi_{l,m,n}^{(p)}(x_{u,n}) \) are ob-
tained using message filtering (see Section III-B1). For each \( m \in \mathcal{M}_{i,n} \), \( \hat{\sigma}^2_{i,m} \) is calculated. For each \( m \in \mathcal{T} \setminus \mathcal{M}_{i,n} \), \( \hat{\sigma}^2_{i,m} \) is formally set to \( \infty \).

**Step 2 (min-consensus algorithm):** For each \( m \in \mathcal{T} \):

a) Initialization: \( \zeta_i^{(0)} = \left\{ \left( \hat{x}_{m,1}^{(j)}, \hat{\sigma}^2_{i,m} \right) \right\}_{j=1}^{\infty} \) if \( m \in \mathcal{M}_{i,n} \) and \( \zeta_i^{(0)} = (\emptyset, \infty) \) if \( m \in \mathcal{T} \setminus \mathcal{M}_{i,n} \).

b) For \( i = 1, 2, \ldots, I \), where \( I \) is the diameter of the network formed by all CAs (or by all CAs in \( S_{m,n} \), if LDT is employed):

First, \( \zeta_i^{(t-1)} \) is broadcast to each CA \( l' \in \mathcal{C}_i \), and \( \zeta_i^{(t-1)} \) is received from each CA \( l' \in \mathcal{C}_i \). Then, \( \zeta_i \) is updated by setting

\[
\zeta_i^{(t)} = \zeta_i^{(t-1)}, \quad \text{where } l' = \arg\min_{l' \in \mathcal{C}_i} \hat{\sigma}^2_{i,m}.
\]

The particles \( \left\{ \hat{x}_{m,1}^{(j)} \right\}_{j=1}^{\infty} \) contained in \( \phi_i^{(t)} \) are used as the PR of \( \phi_{i,m}^{(t)} (\hat{x}_{m,1}) \).

We note that only Step 2b requires communication between CAs; the other steps are performed locally at each CA.

### C. Communication Requirements

In the following discussion of the communication requirements of the proposed CoSLAT algorithm, we assume for simplicity that all states \( x_{k,n}, k \in \mathcal{A} \) have identical dimension \( L \) and all \( \tilde{x}_{k,n}, \tilde{k} \in \mathcal{A} \) (i.e., the states actually involved in the measurements, cf. (2) and (3)) have identical dimension \( \tilde{L} \). Furthermore, we denote by \( N \) the number of consensus iterations used in the LC, by \( N' \) the order of the basis expansion used in the LC, by \( P \) the number of message passing iterations, by \( J \) the number of particles, and by \( \tilde{J} \) the network diameter.

- For LC-based calculation of the target beliefs \( b^{(p)} (x_{m,n}) \), \( m \in \mathcal{T} \) (see Section VII-A and Step 2a in Algorithm 2), at each time \( n \), CA \( l \in \mathcal{S} \) broadcasts \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) real values to CAs \( l' \in \mathcal{C}_i \). In the case of LDT (see Section VII-A), \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) is reduced to \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \). Note that in the LDT case \( C \) is smaller, and even much smaller for a large network.

- For calculation of the CA beliefs \( b^{(p)} (x_{l,n}) \) (see Section VII-B and Step 2e in Algorithm 2), at each time \( n \), CA \( l \in \mathcal{S} \) broadcasts \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) real values to CAs \( l' \in \mathcal{C}_i \). In the case of LDT (see Section VII-B), \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) is reduced to \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \). If the alternative processing of Section VII-B is used, then at time \( n = 1 \), CA \( l \in \mathcal{S} \) broadcasts \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) real values to CAs \( l' \in \mathcal{C}_i \) (in addition to \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) and \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \)). In the case of LDT, \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) is reduced to \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \).

- In the case of LDT, if at time \( n \) a target \( m \in \mathcal{T} \) enters the measurement region of CA \( l \in \mathcal{S} \), i.e., \( l \in S_m \setminus S_m, n = 1 \), then \( \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} \) real values are transmitted from one arbitrary CA \( l' \in S_n \cap \mathcal{C}_i \) to CA \( l \).

Therefore, at time \( n \geq 1 \), the total number of real values broadcast by CA \( l \in \mathcal{S} \) during \( P \) message passing iterations is

\[
\tilde{N}^{|\mathcal{T}|} = \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} + \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} = P (\tilde{J} L + \tilde{C} R | \mathcal{T} |).
\]

If the alternative processing is used, then at time \( n = 1 \),

\[
\tilde{N}_1^{|\mathcal{T}|} = \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} + \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} + \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} = P (\tilde{J} L | \mathcal{T} | + \tilde{J} L + \tilde{C} R | \mathcal{T} |).
\]

In the case of LDT, at time \( n \geq 1 \), the number of real values broadcast by CA \( l \in \mathcal{S} \) during \( P \) message passing iterations is

\[
\tilde{N}_1^{|\mathcal{T}|} = \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} + \tilde{N}^{|\mathcal{C}^{|\mathcal{T}|}} = P (\tilde{J} L + \tilde{C} R | \mathcal{T} |).
\]

We note that \( \tilde{N}_1^{|\mathcal{T}|} \) is very small, and even much smaller for a large network.

### VIII. Simulation Results

We will study the performance and communication requirements of the proposed CoSLAT message passing algorithm in a dynamic scenario and in a static scenario.

**A. Dynamic Scenario**

We consider a network of \( |S| = 12 \) CAs and \( |T| = 2 \) targets as depicted in Fig. 6. Eight CAs are mobile and four are static anchors (i.e., CAs with perfect location information). Each CA has a communication range of 50 and attempts to localize itself (except for the anchors) and the two targets. The states of the mobile CAs and targets consist of location and velocity, i.e., \( x_{k,n} = (x_{k,n}, x_{k,n}, \dot{x}_{k,n}, \dot{x}_{k,n})^T \). CAs \( l \in S \) acquire distance measurements according to (4), i.e., \( y_{l,k,n} = \| \tilde{x}_{k,n} - \tilde{x}_{k,n} \| + v_{l,k,n} \), where \( \tilde{x}_{k,n} = (x_{k,n}, x_{k,n}, \dot{x}_{k,n}, \dot{x}_{k,n})^T \) is the location of agent \( k \in \mathcal{A} \) and the measurement noise \( v_{l,k,n} \) is independent across \( l, k, \) and \( n \) and Gaussian with variance \( \sigma^2_n = 2 \). The measurement regions of four of the mobile CAs are initially located near the corners.

The states of the mobile CAs and targets evolve independently according to \( \dot{x}_{k,n} = G x_{k,n-1} + W u_{k,n}, n = 1, 2, \ldots \)
The driving noise vectors \( u_{k,n} \in \mathbb{R}^2 \) are Gaussian, i.e., \( u_{k,n} \sim \mathcal{N}(0, \sigma_k^2 I) \), with variance \( \sigma_k^2 = 5 \cdot 10^{-5} \) for the mobile CAs and \( \sigma_k^2 = 10^{-3} \) for the targets; furthermore, \( u_{k,n} \) and \( u_{k',n'} \) are independent unless \( (k,n) = (k',n') \). Each mobile CA starts moving only when it is sufficiently localized in the sense that the empirical variance of the estimated location vector is below \( 5 \sigma^2 = 10 \); it then attempts to reach the center of the scene, \( \hat{x}_c = (37.5 \ 37.5)^T \), in 75 time steps. The mobile CA trajectories are initialized using a Dirac-shaped prior located at \( \mu_{i,0} = (x_{i1,0}, x_{i2,0})/75 \) \( (\hat{x}_{i1,c}-x_{i1,0})/75 \) \( (\hat{x}_{i2,c}-x_{i2,0})/75 \), where \( x_{i1,0} \) and \( x_{i2,0} \) are chosen as shown in Fig. 4. The two target trajectories are initialized using a Dirac-shaped prior located at \( (20 \ 0.75 \ 1)^T \) and \( (75 \ 20 \ -0.75 \ 1)^T \) (see Fig. 5). Note that knowledge of these locations is not used in our simulations of the various algorithms.

We compare the proposed CoSLAT algorithm and its low-complexity variant according to Section VII-C—briefly termed “CoSLAT-1” and “CoSLAT-2,” respectively—with that of a reference method that separately performs CSL by means of the nonparametric BP method of [10] and DTT by means of the LC-based distributed PF of [15]; the latter uses the (mobile) CA location estimates provided by CSL. In all three methods, \( P = 2 \) message passing iterations and \( J = 1000 \) particles are used; the alternative processing at time \( n = 1 \) described in Section VII-D is employed; and the LC scheme uses a second-order polynomial approximation [15], resulting in an expansion order of \( R = 9 \), and an average consensus [30] with \( C = 6 \) iterations. For our simulations of the algorithms, we used a location prior for targets and mobile CAs that is uniform on \([-200, 200] \times [-200, 200] \) and a Gaussian velocity prior for the mobile CAs (after the mobile CA is sufficiently localized as described above) with mean \((\hat{x}_{i1,c,-}\hat{x}_{i1,l,n})/75 \) \( (\hat{x}_{i2,c,-}\hat{x}_{i2,l,n})/75 \) and covariance matrix diag \((10^{-3} \ 10^{-3}) \). Here, \( \hat{x}_{i,n} \) is the location estimate at the time \( n \) at which mobile CA \( i \) is sufficiently localized for the first time. We note that this scenario cannot be tackled by SLAT algorithms [16]–[20] since it involves mobile CAs.

Fig. 7 shows the simulated root-mean-square errors (RMSEs) of self-localization and target localization for \( n = 1, \ldots, 75 \). The self-localization RMSE was determined by averaging over all mobile CAs and over 100 simulation runs, and the target localization RMSE by averaging over all mobile CAs, all targets, and 100 simulation runs. It is seen that CoSLAT-1 and CoSLAT-2 perform almost equally well, both with respect to self-localization and target tracking; thus, the significantly lower complexity of CoSLAT-2 is offset by only a small performance loss. Furthermore, the self-localization RMSE of both CoSLAT algorithms is significantly smaller than that of the reference method. This is because with pure CSL, the mobile CAs initially located near the corners (termed “corner CAs” in what follows) do not have enough partners for accurate self-localization, whereas with CoSLAT, these CAs can use their measured distances to the targets to calculate the messages from the target nodes, \( \phi_{m,n}(x_{m,n}) \), which support self-localization. Finally, also the target tracking RMSE of both CoSLAT algorithms is significantly smaller than that of the reference method for almost all times. This is because with separate CSL–DTT, the poor self-localization of the corner CAs results in a degraded target tracking performance.

The quantities determining the communication requirements of CoSLAT-1 and CoSLAT-2 according to Section VII-C are as follows. We have \( N_{LC} = 216 \), \( N_{NBPr} = 4000 \), and \( N_{APR} = 24000 \). For the corner CAs \( l \), \( N_{APR,l} = 108 \) at times \( n \) where a target is measured, i.e., \( |M_{l,n}^m| = 1 \), and \( N_{LC} = 0 \) otherwise. For all other CAs, \( N_{LC} = 216 \). Furthermore, \( N_{APR} = 12000 \) for the two lower corner CAs, \( N_{APR} = 24000 \) for the two upper corner CAs, and \( N_{APR} = 12000 \) for all other CAs. The resulting average total communication requirements per CA and time step, averaged over all CAs, all times, and all simulation runs, are \( \bar{N}_{TOT} = 4536 \) and \( \bar{N}_{TOT} = 4360 \). It is seen that \( \bar{N}_{TOT} \) is not much smaller than \( \bar{N}_{TOT} \); however, this can be very different in other scenarios (e.g., in Section VII-D). At the two times where a target enters the measurement region of an upper corner CA, additionally \( N_{AXT} = 4000 \). For the reference method, we have \( N_{ref} = 4268 \) and \( N_{ref} = 4180 \). (Note that for the reference method, \( N_{APR} = N_{APR}/P \), \( N_{APR} = N_{APR}/P \), \( N_{LC} = N_{LC}/P \), and \( N_{LC} = N_{LC}/P \) because the DTT part performs only one message passing iteration.)

In Fig. 8 we show the self-localization and target localization RMSEs averaged over time \( n \) versus the measurement range of the corner CAs. For small and large measurement range, CoSLAT performs similarly to the reference method but for different reasons: When the measurement range is smaller than 12.5, the targets appear in the measurement regions of the corner CAs only with a very small probability. Thus, at most times, the messages \( \phi_{m,n}(x_{m,n}) \) from the target nodes cannot be calculated. For measurement ranges larger than 25, the corner CAs measure three well-localized CAs at time \( n = 1 \), and thus they are also able to localize themselves using pure CSL. For measurement ranges between 14 and 25, CoSLAT significantly outperforms the reference method (cf. our discussion of Fig. 7). It is furthermore seen that up to a measurement range of 25, the target tracking RMSE of

\[
G = \begin{pmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}, \quad W = \begin{pmatrix}
0.5 & 0 \\
0 & 0.5 \\
1 & 0 \\
0 & 1
\end{pmatrix}.
\]
the reference method increases with increasing measurement range. This is because in the reference method, only an estimate of the CA locations is used in DTT. Therefore, the poorly localized corner CAs negatively affect DTT, and this situation becomes more likely with increasing measurement range. By contrast, the target tracking RMSE of CoSLAT stays constant for all measurement ranges. This is because in CoSLAT, the beliefs of the CA locations are used in DTT, i.e., the actual uncertainty about the CA locations is taken into account. Thereby, the effect of poorly localized CAs on the tracking performance is considerably reduced.

### B. Static Scenario

Next, we consider a network of \(|S| = 63\) static CAs and \(|T| = 50\) static targets. 13 CAs are anchors located as depicted in Fig. 9. The 50 remaining CAs and the 50 targets are randomly (uniformly) placed in a field of size 100 × 100; a realization of the locations of the non-anchor CAs and targets is shown in Fig. 9. The states of the non-anchor CAs and of the targets are the locations, i.e., \(x_{k,n} = \hat{x}_{k,n} = (x_{1,k,n}, x_{2,k,n})^T\). Each CA performs distance measurements according to (3) with noise variance \(\sigma_v^2 = 2\). The non-anchor CAs have a measurement range of 20, whereas the anchors acquire measurements in the entire field. The communication range of each CA is 50. The prior for the non-anchor CAs and for the targets is uniform on \([-200, 200] \times [-200, 200]\). In all three methods, \(J = 1000\) particles are used. The LC scheme uses a second-order polynomial approximation, resulting in \(R = 9\), and an average consensus with \(C = 15\) iterations. Since all CAs and targets are static, we simulate only a single time step. This scenario is similar to that considered in [2] for pure CSL, except that 50 of the CAs used in [2] are replaced by targets and also anchor nodes perform measurements. We note that in this static scenario, our CoSLAT framework reduces to a distributed implementation of particle-based SLAT [17].

Fig. 10 shows the overall agent localization RMSE (i.e., the average RMSE of both CA self-localization and target localization) versus the message passing iteration index \(p\). This error was determined by averaging over all agents and over 100 simulation runs. It is seen that CoSLAT-1 and CoSLAT-2 perform almost equally well, with a slightly faster convergence of CoSLAT-1, and significantly better than the reference method. Again, the better performance of CoSLAT is due to the fact that CAs that do not have enough partners for self-localization can use messages \(\hat{p}_{m \rightarrow n}(x_{m,n})\) from well-localized target nodes to better localize themselves.

In this scenario, assuming \(P = 3\), we have \(N_{LC} = 20250\) and \(N_{NPB} = 6000\). The remaining quantities determining the communication requirements depend on the locations of the non-anchor CAs and the targets, which are randomly chosen for each simulation run. The average quantities—averaged over all CAs, all times, and all simulation runs—were obtained as \(N_{LCR} = 4650\), \(N_{AP} = 1.20 \cdot 10^6\), and \(N_{APR} = 3.45 \cdot 10^5\). The average total communication requirements were obtained as \(N_{TOT} = 1.22 \cdot 10^6\) and \(N_{TOTr} = 3.54 \cdot 10^5\). Note that now \(N_{TOTr}\) is significantly smaller than \(N_{TOT}\), due to the relatively large network size. For the reference method, we obtained \(N_{ref} = 4.10 \cdot 10^5\) and \(N_{TOTr} = 1.22 \cdot 10^5\).

### IX. Conclusion

The proposed framework of cooperative simultaneous localization and tracking (CoSLAT) provides a consistent combination of cooperative self-localization (CSL) and distributed target tracking (DTT) for multiple mobile agents and targets. CoSLAT uses pairwise measurements between agents and targets and between agents. Starting from a factor graph formulation of the CoSLAT problem, we developed a particle-based, distributed belief propagation (BP) algorithm for CoSLAT.
This algorithm employs the likelihood consensus scheme for a distributed calculation of the product of the target messages. More generally, the proposed integration of the likelihood consensus in particle-based BP solves the problem of accommodating noncooperative agent nodes in distributed BP implementations. Thus, it may also be useful for other distributed inference problems.

The main advantage of the proposed CoSLAT framework and BP methodology, over both separate CSL and DTT and simultaneous localization and tracking (SLAT), is a probabilistic information transfer between CSL and DTT. This information transfer allows CSL to support DTT and vice versa. Our simulation results demonstrated that this principle can result in significant improvements in both self-localization and target tracking performance compared to state-of-the-art algorithms.

The CoSLAT framework and methodology can be extended to accommodate additional tasks (i.e., in addition to CSL and DTT) that involve local states of cooperative agents and/or global states of noncooperative agents. Examples of such tasks include distributed synchronization [27], [36] and cooperative mapping [37].

REFERENCES

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, “Locating the nodes: Cooperative localization in wireless sensor networks,” IEEE Signal Process. Mag., vol. 22, pp. 54–69, Jul. 2005.

[2] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” Proc. IEEE, vol. 97, pp. 427–450, Feb. 2009.

[3] J. Liu, M. Chu, and J. Reich, “Multitarget tracking in distributed sensor networks,” IEEE Signal Process. Mag., vol. 24, pp. 36–46, May 2007.

[4] H. Aghajan and A. Cavallaro, Multi-Camera Networks: Principles and Applications. Burlington, MA: Academic Press, 2009.

[5] F. Bullo, J. Cortes, and S. Martinez, Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms. Princeton, NJ: Princeton University Press, 2009.

[6] A. Nayak and I. Stojmenovic, Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication. Hoboken, NJ: Wiley, 2010.

[7] P. Corke, T. Wark, R. Jurdiak, W. Hu, P. Valencia, and D. Moore, “Environmental wireless sensor networks,” Proc. IEEE, vol. 98, no. 11, pp. 1903–1917, 2010.

[8] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Wilsky, “Nonparametric belief propagation for self-localization of sensor networks,” IEEE J. Sel. Areas Commun., vol. 23, pp. 809–819, Apr. 2005.

[9] C. Pedersen, T. Pedersen, and B. H. Fleury, “A variational message passing algorithm for sensor self-localization in wireless networks,” in Proc. IEEE ISIT-11, Saint Petersburg, Russia, pp. 2158–2162, Aug. 2011.

[10] J. Lien, J. Fernet, W. Srichavengsup, H. Wymeersch, and M. Z. Win, “A comparison of parametric and sample-based message representation in cooperative localization.” Int. J. Navig. Offsh., 2012.

[11] V. Savic and S. Zazo, “Reducing communication overhead for cooperative localization using nonparametric belief propagation,” IEEE J. Sel. Areas Commun., vol. 29, pp. 1948–1958, Dec. 2011.

[12] T.-D. Pham, H. Q. Ngo, V.-D. Le, S. Lee, and Y.-K. Lee, “Broadcast gossip based distributed hypothesis testing in wireless sensor networks,” in Proc. ATM-09, Haiphong, Vietnam, pp. 84–87, 2009.

[13] G. Dedeoglu and G. S. Sukhatme, “Landmark-based matching algorithms for cooperative mapping with autonomous robots,” in Distributed Autonomous Robotic Systems 4, New York, NY, pp. 251–260, Springer, 2000.