

Bayesian Cooperative Localization Using Received Signal Strength With Unknown Path Loss Exponent: Message Passing Approaches

Di Jin, Feng Yin, Carsten Fritsche, Fredrik Gustafsson and Abdelhak M. Zoubir

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-165498>

N.B.: When citing this work, cite the original publication.

Jin, Di, Yin, F., Fritsche, C., Gustafsson, F., Zoubir, A. M., (2020), Bayesian Cooperative Localization Using Received Signal Strength With Unknown Path Loss Exponent: Message Passing Approaches, *IEEE Transactions on Signal Processing*, 68, 1120-1135. <https://doi.org/10.1109/TSP.2020.2969048>

Original publication available at:

<https://doi.org/10.1109/TSP.2020.2969048>

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/index.html>

©2020 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



Bayesian Cooperative Localization Using Received Signal Strength With Unknown Path Loss Exponent: Message Passing Approaches

Di Jin, Feng Yin, Carsten Fritsche, Fredrik Gustafsson, and Abdelhak M. Zoubir

Abstract—We propose a Bayesian framework for the received-signal-strength-based cooperative localization problem with unknown path loss exponent. Our purpose is to infer the marginal posterior of each unknown parameter: the position or the path loss exponent. This probabilistic inference problem is solved using message passing algorithms that update messages and beliefs iteratively. For numerical tractability, we combine the variable discretization and Monte-Carlo-based numerical approximation schemes. To further improve computational efficiency, we develop an auxiliary importance sampler that updates the beliefs with the help of an auxiliary variable. An important ingredient of the proposed auxiliary importance sampler is the ability to sample from a normalized likelihood function. To this end, we develop a stochastic sampling strategy that mathematically interprets and corrects an existing heuristic strategy. The proposed message passing algorithms are analyzed systematically in terms of computational complexity, demonstrating the computational efficiency of the proposed auxiliary importance sampler. Various simulations are conducted to validate the overall good performance of the proposed algorithms.

Index Terms—Belief propagation, cooperative localization, message passing, received signal strength, stochastic sampling.

I. INTRODUCTION

LOCATION awareness plays a key enabling role in the expanding location-based services and has a wide range of emerging applications, such as medical services [1], intelligent transportation systems [2], Internet of Things [2], autonomous vehicles [3], crowd sensing [4], [5] and smart environment [6]. Cooperative localization [7]–[10], where all internode measurements can be exploited, has many appealing advantages, among others, expanding the capabilities of locating positions without ambiguity and improving the performance on estimation accuracy. The benefits of cooperation among nodes have been theoretically demonstrated in [11], [12]. The existing cooperative localization algorithms can be categorized into deterministic and probabilistic approaches, depending on whether the localization problem is formulated in a probabilistic manner. In the first category, the positions (and model parameters if any) are assumed to be deterministic but unknown, and only a deterministic point estimate is provided for each unknown parameter. Classical approaches, to mention some, include the maximum likelihood (ML)

approach [7], convex-optimization-based algorithms [13]–[17], multidimensional scaling (MDS) [18], [19] and expectation-conditional maximization (ECM) [20]. On the other hand, the Bayesian approaches treat the positions as random variables and formulate cooperative localization as a probabilistic inference problem. These approaches take advantage of prior information of parameters. Most importantly, the posterior distribution of each position is inferred, which contains much more information than just one deterministic point estimate, e.g., the modality of the position and its associated uncertainty. Representative probabilistic approaches include the nonparametric belief propagation (NBP) [21], [22], sum-product algorithm over a network (SPAWN) [8] and their low-complexity variants [23]–[26].

Among different position-related signal metrics, received signal strength (RSS) has gained much attention due to its ubiquitousness in wireless radio frequency signals [27]. For instance, an RSS indicator (RSSI) has been encoded in the IEEE 802.15.4 standards [28]. Despite its comparatively high uncertainty about position, RSS measurement can be exploited to enable low-cost, simple and opportunistic localization systems, without the need of additional hardware. However, many existing works on RSS-based localization, such as [14], [29], are based on the assumption that the classical path loss propagation model is perfectly known. This oversimplified assumption is impractical for two reasons. Firstly, the estimation of these model parameters usually relies on a laborious calibration phase, where a large amount of training data needs to be collected and processed. However, such a calibration step is very time consuming and even impossible in many scenarios, such as monitoring and surveillance applications in hostile or inaccessible environments [30]. Secondly, these model parameters, particularly the path loss exponent (PLE), are time varying, due to the changing environment, e.g., weather conditions or human behaviors [31], [32]. Without a frequent recalibration, the resulting mismatch will significantly deteriorate the localization performance. To overcome this problem, these model parameters should be assumed unknown and jointly estimated with the positions.

In this paper, we focus on the case with an unknown PLE, because a slight deviation of PLE may severely deteriorate the localization performance, as theoretically and algorithmically demonstrated in [33], [34]. For the case of noncooperative localization, there exist several works dealing with unknown PLE. In [30], the target position and the PLE are estimated jointly by solving an ML problem using the Levenberg-

D. Jin and A. M. Zoubir are with the Signal Processing Group at Technische Universität Darmstadt, Darmstadt, Germany. F. Yin (corresponding author) is with the School of Science and Engineering at Chinese University of Hong Kong, Shenzhen, China. C. Fritsche and F. Gustafsson are with the Department of Electrical Engineering at Linköping University, Linköping, Sweden.

Marquardt algorithm. In [34], [35], the ML problem is first relaxed by linearizing the problem and then simplified by replacing the position variable with a function of the PLE variable. By doing so, the cost function depends only on the one-dimensional (1D) PLE variable, and the resulting optimization problem can be readily solved using grid search. In [30], [36], the location is estimated by eliminating the nuisance parameter: the PLE parameter (or several other model parameters). The original ML problem in [30] is simplified by representing the PLE as a function of the position variable in [33]. In [37], along with several model parameters, the location is estimated based on the expectation and maximization criterion. In [38], [39], the location and the PLE are estimated in an alternating manner. More precisely, the position is estimated based on an initialized (or estimated) PLE, and afterwards the PLE is estimated based on the updated position estimate. This procedure iterates until certain termination condition is met. In the cooperative case, RSS-based localization with an unknown PLE is even more challenging. To the best of our knowledge, only very limited works exist, including [15], [17], [39], where the alternating strategy is adopted to handle the unknown PLE, like in the noncooperative case. In our view, despite its straightforwardness and simplicity, such an alternating strategy is quite heuristic and lack of theoretical support.

In contrast to previous works, we treat the PLE as a random variable and formulate the problem in a Bayesian framework. The reasons are as follows. First, when the PLEs between different propagation links differ, a random variable characterizing the averaging behavior of the collection of all PLEs is more suitable than just one deterministic PLE value. Second, characterizing the PLE as a random variable enables us to integrate any prior information, if available, into the parameter estimation. Under the Bayesian umbrella, the cooperative localization problem with an unknown PLE becomes a probabilistic inference problem. In this problem, we derive message passing algorithms to infer the marginalized posterior distribution of each unknown parameter: the position or the PLE. For mathematical tractability, we combine the variable discretization and Monte-Carlo-based numerical approximation mechanisms. In addition, to reduce the computational complexity, we propose an auxiliary importance sampler for belief update that has a complexity order scaling linearly with the number of samples. Moreover, we develop a novel strategy for sampling from a normalized likelihood function, which plays an important role in the auxiliary importance sampler and mathematically interprets and corrects an existing heuristic sampling strategy. The proposed sampling strategy will benefit many existing works, such as [8], [22], since this task is an embedded step in many message-passing-based cooperative localization algorithms.

This paper is organized as follows: In Section II, we formulate the RSS-based cooperative localization problem with an unknown PLE mathematically. Fundamental concepts in message passing algorithms are given in Section III. We discuss how to approximate the messages in Section IV and demonstrate how to update the beliefs approximately in Section V. Some important issues are discussed in Section VI. The proposed algorithms are evaluated using extensive simulations

in Section VII. Finally, Section VIII concludes the paper.

Notation: Throughout this paper, boldface lowercase letter \mathbf{x} is reserved for vectors. $\|\cdot\|$ stands for the Euclidean norm, and $|\cdot|$ denotes the cardinality of a set. $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2 ; $\mathcal{U}[a, b]$ denotes a uniform distribution with boundaries a and b ; $\log\mathcal{N}(\mu, \sigma^2)$ denotes a log-normally distributed random variable x with μ and σ^2 being the mean and variance of $\log x$. $f(\cdot)$ and $p(\cdot)$ are reserved for probability density functions (pdf) and probability mass functions, respectively. $f_{\mathcal{N}}$ denotes the pdf of a Gaussian distribution. $\Gamma \setminus i$ represents a set consisting all elements in the set Γ excluding the element i . $\{x^l\}_{l=1}^L$ symbolizes a collection of samples $\{x^1, \dots, x^L\}$.

II. PROBLEM FORMULATION

Consider a wireless sensor network (WSN) in 2D space with two types of nodes: blindfolded nodes (*agents*) with unknown locations and reference nodes (*anchors*) with known locations. Let $\mathbf{x}_i = [x_i, y_i]^T$ denote the location of each node, where $i \in S_u = \{1, \dots, N_u\}$ for an agent and $i \in S_a = \{N_u + 1, \dots, N\}$ for an anchor. The index set of all nodes is denoted by S , and we have $S = S_u \cup S_a$. Two nodes i and j are neighbors if they are allowed to communicate with each other. We denote the index set of node i 's neighbors by Γ_i .

Using the well-known log-distance path loss propagation model, the RSS measurement r_{ij} , coming from node i and received by node j , is given by

$$r_{ij} = A_i - 10\alpha \log_{10}(d_{ij}/d_0) + v_{ij}, \quad (1)$$

where d_0 is a predefined reference distance. A_i denotes the reference power in dBm at d_0 , and is assumed to be known. α denotes the path loss exponent (PLE) that is assumed to be an unknown random variable and $d_{ij} \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|$ is the Euclidean distance. v_{ij} stands for the log-normal shadowing error that is modeled by $v_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$. We assume measurements to be symmetric, i.e., r_{ij} and r_{ji} . The collection of all RSS measurements is denoted by $\mathbf{r} \triangleq \{r_{ij} : (i, j) \in \Gamma\}$, where (i, j) represents that nodes i and j are neighbors, and $\Gamma \triangleq \{(i, j) : j \in \Gamma_i \text{ and } j > i; i \in S_u\}$ denotes the set of all pairs of neighboring nodes. In line with the majority of the existing works, we assume that these shadowing measurement errors v_{ij} for all $(i, j) \in \Gamma$ are independent. The distribution of v_{ij} , denoted by $f_{v_{ij}}(v_{ij})$, is assumed to be known.

It is important to stress that the propagation environment implied by the measurement model (1) is homogeneous, i.e., all links share a common PLE α . On the other hand, an inhomogeneous environment is defined for links with different PLEs. The same definitions can be found in [40] as well. It is true that each propagation link, e.g., the connection between node i and node j , should have an individual PLE α_{ij} . However, this will lead to an under-determined problem, since besides the unknown locations, there will be one unknown parameter α_{ij} associated with each measurement r_{ij} . As a compromise between model accuracy and feasibility, a homogeneous propagation environment is considered throughout this work. It is noticeable that even for a homogeneous propagation environment, the problem of RSS-based cooperative

localization with an unknown PLE is readily challenging and rarely studied.

From a Bayesian perspective, we treat the PLE α and each position \mathbf{x}_i , $i \in S$, as random variables, whose prior distributions are denoted by $f(\alpha)$ and $f(\mathbf{x}_i)$, $i \in S$, respectively. All positions and the PLE variable are assumed to be mutually independent, i.e., $f(\alpha, \mathbf{x}_1, \dots, \mathbf{x}_N) = f(\alpha) \cdot f(\mathbf{x}_1) \cdots f(\mathbf{x}_N)$. Our purpose is to infer the marginalized posterior distribution (marginal posterior) of each unknown parameter, which is $f(\alpha|\mathbf{r})$ or $f(\mathbf{x}_i|\mathbf{r})$, $i \in S_u$, from the measurements \mathbf{r} and the prior information about all parameters.

III. FUNDAMENTALS ON COOPERATIVE LOCALIZATION VIA MESSAGE PASSING

To infer the marginal posterior of the PLE variable α and that of each position \mathbf{x}_i , $i \in S_u$, we start with the joint posterior distribution $f(\mathbf{x}_1, \dots, \mathbf{x}_N, \alpha | \mathbf{r})$. Under the assumptions made in the preceding section, it has the form of

$$f(\mathbf{x}_1, \dots, \mathbf{x}_N, \alpha | \mathbf{r}) \propto f(\alpha) \prod_{i=1}^N \left(f(\mathbf{x}_i) \prod_{j \in \Gamma_i, j > i} f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) \right). \quad (2)$$

Intuitively, the marginal posterior, e.g., $f(\mathbf{x}_i | \mathbf{r})$, can be calculated as follows:

$$f(\mathbf{x}_i | \mathbf{r}) = \int \cdots \int f(\mathbf{x}_1, \dots, \mathbf{x}_N, \alpha | \mathbf{r}) \, d\mathbf{x}_{1:N \setminus i} d\alpha.$$

However, this is intractable due to the high dimensionality of the problem. A well-know local message passing algorithm, called *belief propagation* (BP), enables approximate marginalization for loopy graphs in an efficient fashion [41].

A. Fundamentals on Belief Propagation

To facilitate the derivation of the proposed message passing algorithms, we start with the fundamentals on belief propagation. For that, a graphical model, *factor graph*, will be introduced first. For simplicity, we ignore the PLE variable α in this subsection. There are two distinctive types of nodes in the factor graph, the position variables in circles and the factors in squares. An example is depicted in Fig. 1. The factors stand for functions, such as the likelihood functions and the prior distribution functions. Two position variables are connected via a factor if there is a measurement between them. The basic operation in BP is the message passing procedure, which can be interpreted as exchanging statistical information on adjacent nodes of the factor graph. According to the message passing rules in BP [41], the message passed from factor f_{ij} to variable node \mathbf{x}_i is obtained as

$$m_{f_{ij} \rightarrow \mathbf{x}_i}^n(\mathbf{x}_i) \propto \int f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j) f(\mathbf{x}_j) \underbrace{\prod_{t \in \Gamma_j \setminus i} m_{f_{tj} \rightarrow \mathbf{x}_j}^{n-1}(\mathbf{x}_j)}_{\tilde{m}(\mathbf{x}_j)} \, d\mathbf{x}_j.$$

Here, the superscript n is the iteration index, and $\Gamma_j \setminus i$ denotes the set of all neighbors of node j excluding node i . Roughly speaking, the message $m_{f_{ij} \rightarrow \mathbf{x}_i}^n(\mathbf{x}_i)$ can be interpreted as the statistical information on position \mathbf{x}_i coming from its neighbor j . This is obtained by transforming the statistical information

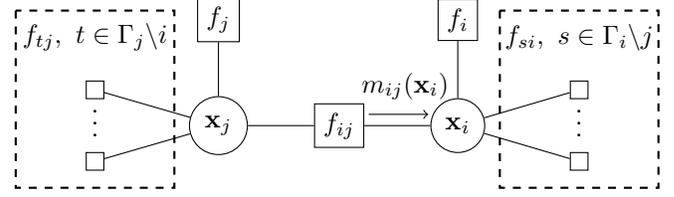


Fig. 1: An illustration of factor graph and belief propagation. For clarity, the overlap between two rectangle blocks are omitted. Here, f_i is the short notation for $f(\mathbf{x}_i)$ and f_{ij} for the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j)$.

($\tilde{m}(\mathbf{x}_j)$) on position \mathbf{x}_j , taking into account the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j)$. Collecting all messages from factors that are connected to position variable \mathbf{x}_i , *belief* at \mathbf{x}_i , denoted by $B^n(\mathbf{x}_i)$, is obtained as

$$B^n(\mathbf{x}_i) \propto f(\mathbf{x}_i) \prod_{j \in \Gamma_i} m_{f_{ij} \rightarrow \mathbf{x}_i}^n(\mathbf{x}_i).$$

The belief $B^n(\mathbf{x}_i)$ can be interpreted as collecting the statistical information on \mathbf{x}_i coming from all its neighbors ($m_{f_{ij} \rightarrow \mathbf{x}_i}^n(\mathbf{x}_i)$, $\forall j \in \Gamma_i$) and its own prior ($f(\mathbf{x}_i)$). Such message passing procedure iterates until certain termination condition is fulfilled. Upon convergence, the belief $B(\mathbf{x}_i)$ is an approximation of the marginal posterior $f(\mathbf{x}_i | \mathbf{r})$.

B. Proposed Message Passing Algorithms

Despite the fact that several works, e.g., [21], [42], exist for cooperative localization via BP, it is impossible to apply them directly to our problem. The reason is that unlike a pairwise potential function in the existing works, the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha)$ in our problem is of order three. This makes the BP algorithm for our problem not straightforward, and, hence, we will derive it explicitly in what follows. We first represent the joint posterior distribution $f(\mathbf{x}_1, \dots, \mathbf{x}_N, \alpha | \mathbf{r})$ using the factor graph, see Fig. 2. The PLE variable is connected to all likelihood functions as it is related to all measurements.

The key idea of the BP is to update a set of messages iteratively, which contribute to calculating the marginal posteriors. With a slight notation abuse, we use f_{ij} as a shorthand notation for the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha)$ from now on. We denote the message from factor f_{ij} to variable α by $m_{f_{ij} \rightarrow \alpha}(\alpha)$ and that from f_{ij} to \mathbf{x}_i by $m_{f_{ij} \rightarrow \mathbf{x}_i}(\mathbf{x}_i)$. The messages $m_{f_{ij} \rightarrow \alpha}(\alpha)$ and $m_{f_{ij} \rightarrow \mathbf{x}_i}(\mathbf{x}_i)$ are updated according to the following rules:

$$m_{f_{ij} \rightarrow \alpha}^n(\alpha) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) f(\mathbf{x}_i) \prod_{s \in \Gamma_i \setminus j} m_{f_{is} \rightarrow \mathbf{x}_i}^{n-1}(\mathbf{x}_i) \cdot f(\mathbf{x}_j) \prod_{t \in \Gamma_j \setminus i} m_{f_{tj} \rightarrow \mathbf{x}_j}^{n-1}(\mathbf{x}_j) \, d\mathbf{x}_i \, d\mathbf{x}_j, \quad (3a)$$

$$m_{f_{ij} \rightarrow \mathbf{x}_i}^n(\mathbf{x}_i) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) f(\mathbf{x}_j) \prod_{t \in \Gamma_j \setminus i} m_{f_{tj} \rightarrow \mathbf{x}_j}^{n-1}(\mathbf{x}_j) \cdot f(\alpha) \prod_{(u,z) \in \Gamma \setminus (i,j)} m_{f_{uz} \rightarrow \alpha}^{n-1}(\alpha) \, d\mathbf{x}_j \, d\alpha. \quad (3b)$$

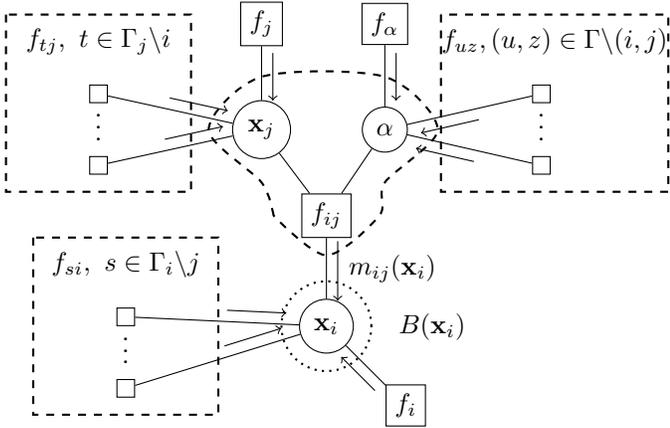


Fig. 2: An illustration of factor graph and belief propagation associated with (2). For clarity, the overlap between three rectangular blocks are omitted. Here, f_i and f_α are short notations for $f(\mathbf{x}_i)$ and $f(\alpha)$, respectively, and f_{ij} for the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha)$.

Here, $\Gamma \setminus (i, j)$ denotes the set of all pairs of connections excluding (i, j) . To facilitate compact notation, we will simplify $m_{f_{ij} \rightarrow \alpha}(\alpha)$ and $m_{f_{ij} \rightarrow \mathbf{x}_i}(\mathbf{x}_i)$ to $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$, respectively. An illustrative explanation of (3b) is depicted in Fig. 2, where the messages enclosed in the dashed circle contribute to calculating the message $m_{ij}(\mathbf{x}_i)$. At the first glance, the message update rule in (3) seems tedious. In the subsequent context, a reformulation of (3) will be given in (5), therewith facilitating the interpretation of the messages $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$. Based on these messages, the beliefs are updated by performing

$$B^n(\alpha) \propto f(\alpha) \prod_{(i,j) \in \Gamma} m_{ij}^n(\alpha), \quad (4a)$$

$$B^n(\mathbf{x}_i) \propto f(\mathbf{x}_i) \prod_{j \in \Gamma_i} m_{ij}^n(\mathbf{x}_i). \quad (4b)$$

Here, $B^n(\alpha)$ and $B^n(\mathbf{x}_i)$ denote the belief of the PLE variable α and the belief of the position variable \mathbf{x}_i in the n -th iteration, respectively. The belief update rule in (4), such as $B^n(\mathbf{x}_i)$, can be interpreted as multiplying the messages coming from all factors connected to \mathbf{x}_i . As an illustrative example, the belief update rule for $B(\mathbf{x}_i)$ is depicted in Fig. 2, where the messages enclosed in the dotted circle contribute to updating $B(\mathbf{x}_i)$.

Comparing (3) with (4), it is obvious that certain terms in (3) can be replaced by (4). By doing so, the message update rule in (3) can be equivalently rewritten into a simpler form, namely,

$$m_{ij}^n(\alpha) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) \frac{B^{n-1}(\mathbf{x}_i) B^{n-1}(\mathbf{x}_j)}{m_{ij}^{n-1}(\mathbf{x}_i) m_{ij}^{n-1}(\mathbf{x}_j)} d\mathbf{x}_i d\mathbf{x}_j, \quad (5a)$$

$$m_{ij}^n(\mathbf{x}_i) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) \frac{B^{n-1}(\mathbf{x}_j) B^{n-1}(\alpha)}{m_{ij}^{n-1}(\mathbf{x}_j) m_{ij}^{n-1}(\alpha)} d\mathbf{x}_j d\alpha. \quad (5b)$$

Such a reformulation results in the succinct message update rules, and the underlying meaning of the messages becomes better revealed in (5). Taking $m_{ij}^n(\mathbf{x}_i)$ as an example, it

implies that certain information on \mathbf{x}_i can be inferred from the likelihood function $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha)$, given the beliefs of \mathbf{x}_j and α . Alternatively, following the idea in [8], the messages can be approximated by ignoring the denominator terms in (5), giving rise to the following message update rules:

$$m_{ij}^n(\alpha) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) B^{n-1}(\mathbf{x}_i) B^{n-1}(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j, \quad (6a)$$

$$m_{ij}^n(\mathbf{x}_i) \propto \iint f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j, \alpha) B^{n-1}(\mathbf{x}_j) B^{n-1}(\alpha) d\mathbf{x}_j d\alpha. \quad (6b)$$

In this paper, the message passing algorithm in light of (4)-(5) is referred to as the BP; while that in light of (4) and (6) is referred to as the SPAWN. Note that the difference between the BP and the SPAWN lies in the message update rule. As will be shown later in Section VI-A, the SPAWN message update rule according to (6) achieves a significant reduction in computational complexity, compared to the BP.

Clearly, the gist of the message passing algorithms is to perform two steps iteratively: updating the messages according to (5) (or (6)) and updating the beliefs according to (4). To give an overview, we summarize the resulting framework for inferring the marginal posteriors $f(\alpha | \mathbf{r})$ and $f(\mathbf{x}_i | \mathbf{r})$, $i \in S_u$, in Algorithm 1. First, we initialize the beliefs, $B^0(\alpha)$ and $B^0(\mathbf{x}_i)$, $i \in S$. Here, one sensible choice for the initial beliefs is their prior distributions. In the n -th iteration, the messages $m_{ij}^n(\alpha)$ and $m_{ij}^n(\mathbf{x}_i)$ are updated using Algorithms 2 and 3, respectively, that will be given in Section IV. Then, the belief of each position, i.e., $B^n(\mathbf{x}_i)$, $i \in S_u$, is updated, either using an importance sampler or using Algorithm 4, to be given in Section V. Finally, the belief $B^n(\alpha)$ is updated, which will be discussed in Section V as well. These operations iterate until certain termination condition is met, for instance, when the maximal number of iterations N_{\max} is arrived. Different from the existing works, this Bayesian framework treats both α and \mathbf{x}_i , $i \in S$, as random variables. It has the advantage that any prior knowledge on α and \mathbf{x}_i , $i \in S$, can be integrated. By doing so, $f(\alpha)$ reflecting the prior knowledge on any particular environment and $f(\mathbf{x}_i)$ representing the prior knowledge of any degree can be exploited. For instance, an anchor with imperfect position information can be easily handled in this framework. Moreover, this framework provides marginal posterior estimate for each unknown parameter, which contains much more information than just one point estimate.

The main challenge in the proposed message passing algorithms lies in that there is no closed-form solution except for two special cases: case with discrete-valued variables and case with jointly Gaussian-distributed continuous-valued variables [43]. In our problem, where the variables are continuous-valued but not jointly Gaussian distributed, we must resort to numerical approximation mechanisms. One naive and simple numerical approximation scheme is to define a set of grid points, on which the beliefs and messages are evaluated. There are two limitations in this approach. First, the number of the grid points grows exponentially with the dimensionality of the variable. Second, for a certain fixed granularity, the number of the grid points along one dimension grows linearly with

Algorithm 1 Cooperative Localization Algorithms

```

1: Initialization:  $B^0(\alpha)$  and  $B^0(\mathbf{x}_i)$  for all  $i \in S$ 
2: for  $n = 1 : N_{\max}$ 
3:   for each  $i \in S_u$ 
4:     for each  $j \in \Gamma_i$ 
5:       if  $j > i$ , then calculate  $m_{ij}^n(\alpha)$ , see Algorithm 2
6:       compute  $m_{ij}^n(\mathbf{x}_i)$ , see Algorithm 3
7:     end for
8:     update and broadcast  $B^n(\mathbf{x}_i)$ , see the importance
       sampler in Section V-A or Algorithm 4
9:   end for
10:  calculate  $B^n(\alpha)$  using (12)
11: end for
  
```

its supported interval. Therefore, this approach is appropriate only when the variable is of low dimensionality and defined on a bounded interval, for instance, the 1D PLE variable α varying in the range of [1.5, 6] [32]. Alternatively, Monte-Carlo-based numerical approximation approaches have been proposed in [21], [43], where both the beliefs and the messages are approximated based on a set of weighted samples. These samples are generated using certain stochastic methods, for instance, Markov Chain Monte Carlo methods in [21], [43]. These sample-based approaches provide an alternative to deal with high-dimensional variables or variables with infinite or relatively large support, such as the position variable \mathbf{x}_i , $i \in S_u$. Taking all the above into consideration, we will discretize α and sample \mathbf{x}_i , $i \in S_u$, using stochastic sampling methods. More specifically, the messages and the belief of α , e.g., $m_{ij}(\alpha)$ and $B(\alpha)$, are only evaluated on a set of predefined grid points $\{\alpha_d^r\}_{r=1}^R$; while the messages and the beliefs of positions, e.g., $m_{ij}(\mathbf{x}_i)$ and $B(\mathbf{x}_i)$, are approximated based on weighted samples. In the next two sections, we will detail the approximation mechanisms for message updating and belief updating.

IV. UPDATING MESSAGES OF POSITIONS AND PLE

In this section, we consider how to update the messages $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$ approximately. We proceed with the BP message update rule, and message updating using the SPAWN can be derived in analogy with the BP. For the moment, we assume that $\{\mathbf{x}_i^{l,n-1}\}_{l=1}^L$, $\{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$ and $\{B(\alpha_d^r)^{n-1}\}_{r=1}^R$ are available, which are the equally weighted samples of $B^{n-1}(\mathbf{x}_i)$, those of $B^{n-1}(\mathbf{x}_j)$ and the evaluation values of $B^{n-1}(\alpha)$ at $\{\alpha_d^r\}_{r=1}^R$, respectively.

The message $m_{ij}^n(\alpha)$ can be updated by approximating the double integral in (5a) using importance sampling [44], giving rise to,

$$m_{ij}^n(\alpha) \propto \sum_{l=1}^L w_{ij \rightarrow \alpha}^{l,n} f(r_{ij} | \mathbf{x}_i^l, \mathbf{x}_j^l, \alpha),$$

$$w_{ij \rightarrow \alpha}^{l,n} \propto \frac{B^{n-1}(\mathbf{x}_i^l) \cdot B^{n-1}(\mathbf{x}_j^l)}{m_{ij}^{n-1}(\mathbf{x}_i^l) \cdot m_{ij}^{n-1}(\mathbf{x}_j^l) \cdot q(\mathbf{x}_i^l, \mathbf{x}_j^l)},$$

where $\{\mathbf{x}_i^l, \mathbf{x}_j^l\}_{l=1}^L$ are samples from the proposal distribution $q(\mathbf{x}_i, \mathbf{x}_j)$, and $w_{ij \rightarrow \alpha}^{l,n}$ is the importance weight satisfying $\sum_{l=1}^L w_{ij \rightarrow \alpha}^{l,n} = 1$. Based on the fact that \mathbf{x}_i and \mathbf{x}_j are decoupled in the non-normalized target distribution $\frac{B^{n-1}(\mathbf{x}_i)}{m_{ij}^{n-1}(\mathbf{x}_i)}$. $\frac{B^{n-1}(\mathbf{x}_j)}{m_{ij}^{n-1}(\mathbf{x}_j)}$, we decouple \mathbf{x}_i and \mathbf{x}_j in the proposal distribution, resulting in $q(\mathbf{x}_i, \mathbf{x}_j) = q(\mathbf{x}_i) \cdot q(\mathbf{x}_j)$. The question that remains to answer is how to choose $q(\mathbf{x}_i)$ and $q(\mathbf{x}_j)$. A sensible choice for $q(\mathbf{x}_i)$ is the belief $B^{n-1}(\mathbf{x}_i)$. The reasons are twofold. First, the belief $B^{n-1}(\mathbf{x}_i)$ is part of \mathbf{x}_i 's non-normalized target distribution $B^{n-1}(\mathbf{x}_i)/m_{ij}^{n-1}(\mathbf{x}_i)$, and it approximates $f(\mathbf{x}_i | \mathbf{r})$. Second, the samples from $B^{n-1}(\mathbf{x}_i)$ are available, and no extra effort is need. For the same reasons, $B^{n-1}(\mathbf{x}_j)$ is chosen as the proposal distribution $q(\mathbf{x}_j)$. Consequently, $m_{ij}^n(\alpha)$ is approximated to

$$m_{ij}^n(\alpha) \propto \sum_{l=1}^L w_{ij \rightarrow \alpha}^{l,n} f(r_{ij} | \mathbf{x}_i^{l,n-1}, \mathbf{x}_j^{l,n-1}, \alpha), \quad (7a)$$

$$w_{ij \rightarrow \alpha}^{l,n} \propto \frac{1}{m_{ij}^{n-1}(\mathbf{x}_i^{l,n-1}) \cdot m_{ij}^{n-1}(\mathbf{x}_j^{l,n-1})}. \quad (7b)$$

Here, $\{\mathbf{x}_i^{l,n-1}\}_{l=1}^L$ and $\{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$ are samples from $B^{n-1}(\mathbf{x}_i)$ and $B^{n-1}(\mathbf{x}_j)$, respectively, and the importance weights fulfill $\sum_{l=1}^L w_{ij \rightarrow \alpha}^{l,n} = 1$. Since we have defined the grid points $\{\alpha_d^r\}_{r=1}^R$, as a last step, $m_{ij}^n(\alpha)$ is evaluated at $\{\alpha_d^r\}_{r=1}^R$. As will be seen later in Section V, evaluating the messages of α at $\{\alpha_d^r\}_{r=1}^R$ can facilitate updating the belief $B(\alpha)$ significantly.

For $m_{ij}^n(\mathbf{x}_i)$ in (5b), we can directly combine the discretization approximation and the importance sampling technique, leading to

$$m_{ij}^n(\mathbf{x}_i) \propto \sum_{r=1}^R \sum_{l=1}^L w_{ij \rightarrow \mathbf{x}_i}^{l,n} \frac{B^{n-1}(\alpha_d^r)}{m_{ij}^{n-1}(\alpha_d^r)} f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha_d^r), \quad (8a)$$

$$w_{ij \rightarrow \mathbf{x}_i}^{l,n} \propto 1/m_{ij}^{n-1}(\mathbf{x}_j^{l,n-1}), \quad (8b)$$

where $\{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$ denote the samples from $B^{n-1}(\mathbf{x}_j)$, and the importance weights fulfill $\sum_{l=1}^L w_{ij \rightarrow \mathbf{x}_i}^{l,n} = 1$. In contrast to $m_{ij}^n(\alpha)$ in (7a), the double-integral problem for $m_{ij}^n(\mathbf{x}_i)$ becomes a double summation in (8a). As compared to $m_{ij}^n(\alpha)$, $m_{ij}^n(\mathbf{x}_i)$ is computationally heavier, incurring that updating the belief $B(\mathbf{x}_i)$ will be computationally intensive as well. In order to approximate $m_{ij}^n(\mathbf{x}_i)$ computationally more efficiently, we treat α in the same manner as \mathbf{x}_j and perform importance sampling for both α and \mathbf{x}_j . However, α can be drawn only from the set of the grid points $\{\alpha_d^r\}_{r=1}^R$, since its non-normalized target distribution $B^{n-1}(\alpha)/M_{ij}^{n-1}(\alpha)$ are evaluated at $\{\alpha_d^r\}_{r=1}^R$. Subsequently, $m_{ij}^n(\mathbf{x}_i)$ can be approximated to

$$m_{ij}^n(\mathbf{x}_i) \propto \sum_{l=1}^L w_{ij \rightarrow \mathbf{x}_i}^{l,n} f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}), \quad (9a)$$

$$w_{ij \rightarrow \mathbf{x}_i}^{l,n} \propto \frac{1}{m_{ij}^{n-1}(\mathbf{x}_j^{l,n-1}) \cdot m_{ij}^{n-1}(\alpha^{l,n-1})}, \quad (9b)$$

where $\{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$ and $\{\alpha^{l,n-1}\}_{l=1}^L$ denote the samples from $B^{n-1}(\mathbf{x}_j)$ and $B^{n-1}(\alpha)$, respectively, and the importance weights fulfill $\sum_{l=1}^L w_{ij \rightarrow \mathbf{x}_i}^{l,n} = 1$. Thank to the additional sampling process, the double summation in (8a) is simplified to a single summation in (9a).

Next, we further transform the message $m_{ij}^n(\mathbf{x}_i)$ to

$$m_{ij}^n(\mathbf{x}_i) \propto \sum_{l=1}^L \tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n} \tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}), \quad (10a)$$

where $\tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1})$ is the normalized likelihood function, as given by

$$\tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}) = Z_{ij}^{-1} f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}), \quad (10b)$$

$$Z_{ij} = \int f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}) d\mathbf{x}_i, \quad (10c)$$

and the mixture weight $\tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n}$ is given by

$$\tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n} \propto Z_{ij} \cdot w_{ij \rightarrow \mathbf{x}_i}^{l,n}, \quad (10d)$$

that satisfies $0 \leq \tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n} \leq 1$ and $\sum_{l=1}^L \tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n} = 1$. The integral in (10c) can be evaluated analytically with the details given in Appendix A. It is noteworthy that (10a) differs from (9a) in that the mixture component $\tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^l, \alpha^l)$ is a normalized likelihood function of \mathbf{x}_i , satisfying the properties of a probability density function (pdf), while $f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^l, \alpha^l)$ in (9a) not. As will be seen later in Section V, $m_{ij}^n(\mathbf{x}_i)$ in the form of (10a) is more advantageous than that in (9a), since it enables the development of an efficient sampling procedure for updating $B(\mathbf{x}_i)$. Finally, Algorithms 2 and 3 summarize the steps for updating the messages, $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$, respectively.

Remark. In the SPAWN, the messages are updated in the same manner as the procedures above. The only difference is that the importance weights in (7b)-(9b) are replaced by

$$w_{ij \rightarrow \alpha}^{l,n} = 1/L, \quad (11a)$$

$$w_{ij \rightarrow \mathbf{x}_i}^{l,n} = 1/L. \quad (11b)$$

Algorithm 2 Message Update of $m_{ij}^n(\alpha)$

- 1: **Input:** $B^{n-1}(\mathbf{x}_i) \triangleq \{\mathbf{x}_i^{l,n-1}\}_{l=1}^L$
 $B^{n-1}(\mathbf{x}_j) \triangleq \{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$
 - 2: **Output:** $\{m_{ij}^n(\alpha_d^r)\}_{r=1}^R$
 - 3: calculate $w_{ij \rightarrow \alpha}^{l,n}$ using (7b) \leftarrow BP
or (11a) \leftarrow SPAWN
 - 4: evaluate $m_{ij}^n(\alpha)$ at $\{\alpha_d^r\}_{r=1}^R$ using (7a)
-

V. UPDATING BELIEFS OF POSITIONS AND PLE

In this section, we will discuss the numerical approximation mechanism for updating the beliefs: $B(\alpha)$ and $B(\mathbf{x}_i)$, $i \in S_u$. First, we consider how to update the belief $B^n(\alpha)$ according to the update rule in (4a). For the reason that $\{m_{ij}^n(\alpha_d^r)\}_{r=1}^R$

Algorithm 3 Message Update of $m_{ij}^n(\mathbf{x}_i)$

- 1: **Input:** $B^{n-1}(\alpha) \triangleq \{\alpha_d^r\}_{r=1}^R$
 $B^{n-1}(\mathbf{x}_j) \triangleq \{\mathbf{x}_j^{l,n-1}\}_{l=1}^L$
 - 2: **Output:** $\{\tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n}, \tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1})\}_{l=1}^L$
 - 3: draw $\alpha^{l,n-1} \sim B^{n-1}(\alpha)$
 - 4: calculate $w_{ij \rightarrow \mathbf{x}_i}^{l,n}$ using (9b) \leftarrow BP
or (11b) \leftarrow SPAWN
 - 5: compute $\tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n}$ and $\tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1})$ using (10)
-

are available for each pair of connection $(i, j) \in \Gamma$, $B^n(\alpha)$ can be readily evaluated at $\{\alpha_d^r\}_{r=1}^R$,

$$B^n(\alpha_d^r) \propto f_\alpha(\alpha_d^r) \prod_{(i,j) \in \Gamma} m_{ij}^n(\alpha_d^r). \quad (12)$$

Thanks to the discretization, updating $B^n(\alpha)$ can be conducted by simply multiplying $|\Gamma|$ real-valued numbers at R grid points.

Next, we consider how to update the beliefs of position variables, for instance $B^n(\mathbf{x}_i)$. By combining (4b)-(10a), we obtain $B^n(\mathbf{x}_i)$ in the form of

$$B^n(\mathbf{x}_i) \propto f(\mathbf{x}_i) \prod_{j \in \Gamma_i} \left(\sum_{l=1}^L \tilde{w}_{ij \rightarrow \mathbf{x}_i}^{l,n} \tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1}) \right). \quad (13)$$

Our purpose is to conduct efficient sampling, i.e., $\mathbf{x}_i \sim B^n(\mathbf{x}_i)$. Here, the target distribution $B^n(\mathbf{x}_i)$ is a product of $|\Gamma_i|$ mixtures, each being a sum of L weighted normalized likelihood functions. Note that the component $\tilde{f}(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^{l,n-1}, \alpha^{l,n-1})$ is in general non-Gaussian. Therefore, updating $B(\mathbf{x}_i)$ boils down to sampling from a product of non-Gaussian mixtures. For notational convenience, we simplify (13) to

$$B(\mathbf{x}) \propto f(\mathbf{x}) \prod_{j=1}^J M_j(\mathbf{x}), \quad M_j(\mathbf{x}) = \sum_{l=1}^L \nu_j^l f_j^l(\mathbf{x}). \quad (14)$$

One straightforward sampling strategy is to construct all components explicitly and to sample from them. This is, however, computationally prohibitive, since the product of J mixtures, each containing L components, is itself a mixture of L^J components. Besides, there exist several samplers in the existing works, including the Gibbs sampler [21] and its related multi-scale sampling strategies in [45], [46]. These approaches, however, require a prerequisite that each $M_j(\mathbf{x})$ is a Gaussian mixture, and, therefore, they are not applicable to our problem. In what follows, we will first revisit an existing sampling approach and then propose an alternative sampler, which has a significantly reduced computational complexity.

A. Importance Sampling as Baseline

First, we consider the technique of importance sampling. The samples and the associated weights are obtained as follows:

$$\mathbf{x}^l \sim q(\mathbf{x}), \quad w^l \propto B(\mathbf{x}^l)/q(\mathbf{x}^l), \quad (15)$$

where $q(\mathbf{x})$ is an appropriate proposal distribution, and the importance weight w^l satisfies $\sum_{l=1}^L w^l = 1$. The possible choices for $q(\mathbf{x})$ are the prior distribution $f(\mathbf{x})$, an evenly weighted sum of J mixtures $\sum_{j=1}^J J^{-1} M_j(\mathbf{x})$ [22] and the message with the smallest entropy, e.g., $M_j(\mathbf{x})$, [23]. The resulting Algorithm 1 with the beliefs updated using the importance sampler in (15) is referred to as BP-IS or SPAWN-IS, for that the messages are updated according to the BP or the SPAWN, respectively. One shortcoming of the importance sampler lies in the high computational load, since computing these L weights $\{w^l\}_{l=1}^L$ according to (15) requires operations of order $\mathcal{O}(JL^2)$ [23]. In order to reduce the computational load, we propose an alternative sampler in what follows.

B. Proposed Auxiliary Importance Sampler

Motivated by [47], we develop an efficient sampler, named as *auxiliary importance sampler* (AIS), for the sampling problem $\mathbf{x} \sim B(\mathbf{x})$. The key idea is to introduce an auxiliary variable ψ_j to each mixture $M_j(\mathbf{x})$. The auxiliary variable ψ_j plays the role of a component label indicator, indicating which component is drawn from the mixture $M_j(\mathbf{x}) = \sum_{l=1}^L \nu_j^l f_j^l(\mathbf{x})$, and it can take value $\psi_j = \kappa$, where $\kappa \in \{1, \dots, L\}$. For instance, if we have $\psi_j = \kappa$, it denotes that the κ -th component $\nu_j^\kappa f_j^\kappa(\mathbf{x})$ is drawn from the mixture $M_j(\mathbf{x}) = \sum_{l=1}^L \nu_j^l f_j^l(\mathbf{x})$. Stacking all J auxiliary variables into a vector, we have the compact auxiliary variable $\boldsymbol{\psi} = [\psi_1, \dots, \psi_J]^T$.

With the help of the auxiliary variable $\boldsymbol{\psi}$, the sampling task $\mathbf{x} \sim B(\mathbf{x})$ can be achieved in two steps:

- 1) Draw $\boldsymbol{\psi}^l \sim p(\boldsymbol{\psi})$,

$$p(\boldsymbol{\psi}) = \int f(\mathbf{x}, \boldsymbol{\psi}) d\mathbf{x} = Z_1^{-1} \int \prod_{j=1}^J \nu_j^{\psi_j} f_j^{\psi_j}(\mathbf{x}) d\mathbf{x}; \quad (16)$$

- 2) Draw $\mathbf{x}^l \sim f(\mathbf{x}|\boldsymbol{\psi}^l)$, conditional on $\boldsymbol{\psi}^l$,

$$f(\mathbf{x}|\boldsymbol{\psi}^l) = Z_2^{-1} \prod_{j=1}^J f_j^{\psi_j^l}(\mathbf{x}). \quad (17)$$

Here, Z_1 and Z_2 are two normalization constants. Neglecting the auxiliary variable samples $\{\boldsymbol{\psi}^l\}_{l=1}^L$, the samples $\{\mathbf{x}^l\}_{l=1}^L$ generated in such a two-step procedure follow the distribution in (14). However, when directly sampling from $p(\boldsymbol{\psi})$ and $f(\mathbf{x}|\boldsymbol{\psi}^l)$ is impossible, as in our case, we can generate samples from two proposal distributions $q(\boldsymbol{\psi})$ and $q(\mathbf{x}|\boldsymbol{\psi}^l)$ and assign certain importance weights to them. This gives rise to the following three-step procedure:

- 1) Draw $\boldsymbol{\psi}^l \sim q(\boldsymbol{\psi})$;
- 2) Draw $\mathbf{x}^l \sim q(\mathbf{x}|\boldsymbol{\psi}^l)$, conditional on $\boldsymbol{\psi}^l$;
- 3) Calculate the importance weight w^l

$$w^l \propto \frac{f(\mathbf{x}^l, \boldsymbol{\psi}^l)}{q(\mathbf{x}^l, \boldsymbol{\psi}^l)} = \frac{f(\mathbf{x}^l, \boldsymbol{\psi}^l)}{q(\boldsymbol{\psi}^l) \cdot q(\mathbf{x}^l|\boldsymbol{\psi}^l)}$$

with the non-normalized joint distribution $f(\mathbf{x}^l, \boldsymbol{\psi}^l)$ given by

$$f(\mathbf{x}^l, \boldsymbol{\psi}^l) = f(\mathbf{x}^l) \prod_{j=1}^J \nu_j^{\psi_j^l} f_j^{\psi_j^l}(\mathbf{x}^l).$$

Up to this point, the problem remained is how to design $q(\boldsymbol{\psi})$ and $q(\mathbf{x}|\boldsymbol{\psi}^l)$, which will be addressed in what follows.

Remark. Note that the underlying condition in the AIS is that the target distribution $B(\mathbf{x})$ must be a product of several mixtures, each being a sum of multiple weighted pdfs. Thanks to the additional message transformation in (10), the belief in (13) satisfies the properties of this condition, meaning that the message transformation in (10) is a prerequisite for the development of the AIS.

1) *Auxiliary Variable $\boldsymbol{\psi}$* : First, we focus on designing an appropriate proposal distribution $q(\boldsymbol{\psi})$. Ideally, $q(\boldsymbol{\psi})$ should resemble the corresponding target distribution $p(\boldsymbol{\psi})$ as closely as possible, and, at the same time, it should be feasible to draw samples from it. To this end, we first recover the original form of the target distribution $p(\boldsymbol{\psi})$. This can be readily achieved by replacing \mathbf{x} , $\nu_j^{\psi_j}$ and $f_j^{\psi_j}(\mathbf{x})$ in (16) with \mathbf{x}_i , $\tilde{w}_{ij \rightarrow \mathbf{x}_i}^{\psi_j}$ and $\tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{\psi_j}, \alpha^{\psi_j})$, respectively, giving rise to

$$p(\boldsymbol{\psi}) \propto \int \prod_{j \in \Gamma_i} \tilde{w}_{ij \rightarrow \mathbf{x}_i}^{\psi_j} \tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{\psi_j}, \alpha^{\psi_j}) d\mathbf{x}_i.$$

To ensure mathematical tractability, we assume that all auxiliary variables in $\{\psi_j : j \in \Gamma_i\}$ are independent, giving rise to $q(\boldsymbol{\psi}) = \prod_{j \in \Gamma_i} q(\psi_j)$ with

$$q(\psi_j = \kappa) = \tilde{w}_{ij \rightarrow \mathbf{x}_i}^\kappa \int \tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^\kappa, \alpha^\kappa) d\mathbf{x}_i = \tilde{w}_{ij \rightarrow \mathbf{x}_i}^\kappa,$$

where the second equality follows from (10b)-(10c).

2) *Position Variable \mathbf{x}* : In order to design $q(\mathbf{x}|\boldsymbol{\psi}^l)$, again, we recover the original form of $f(\mathbf{x}|\boldsymbol{\psi}^l)$. This is done by replacing \mathbf{x} and $f_j^{\psi_j^l}(\mathbf{x})$ in (17) with \mathbf{x}_i and $\tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l})$, respectively, giving rise to

$$f(\mathbf{x}_i|\boldsymbol{\psi}^l) \propto \prod_{j \in \Gamma_i} \tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}). \quad (18)$$

To capture each mixture component in (18), we design $q(\mathbf{x}_i|\boldsymbol{\psi}^l)$ in the form of

$$q(\mathbf{x}_i|\boldsymbol{\psi}^l) = \sum_{j \in \Gamma_i} |\Gamma_i|^{-1} q(\mathbf{x}_i|\mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}, r_{ij}), \quad (19)$$

where $q(\mathbf{x}_i|\mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}, r_{ij})$ should resemble $\tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l})$ as closely as possible, and, at the same time, drawing samples from it remains feasible. For notational clarity, we will replace ψ_j^l with l' , thereby simplifying $q(\mathbf{x}_i|\mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}, r_{ij})$ to $q(\mathbf{x}_i|\mathbf{x}_j^{l'}, \alpha^{l'}, r_{ij})$.

Next, we proceed with designing the proposal distribution $q(\mathbf{x}_i|\mathbf{x}_j^{l'}, \alpha^{l'}, r_{ij})$ for the target distribution $\tilde{f}(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{l'}, \alpha^{l'})$, which is the normalized likelihood function $f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{l'}, \alpha^{l'}) / \int f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^{l'}, \alpha^{l'}) d\mathbf{x}_i$. This task is actually an embedded step in many other works, for instance, under different measurement models in [8], [22], [23], [25], [48]. Therefore, instead of being specific, we generalize this sampling problem to a generic measurement model, given by

$$r_{ij} = h(d_{ij}) + v, \quad v \sim f_v(v). \quad (20)$$

Here r_{ij} denotes any distance-related measurement, $h(d_{ij})$ is a function of the internode distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, and v is an additive measurement error. Our purpose is to sample from the normalized likelihood function, namely,

$$\mathbf{x}_i^l \sim Z^{-1} f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^l), \quad (21)$$

where \mathbf{x}_j^l is a reference position, and Z is a normalization constant, to be precise, $Z = \int f(r_{ij} | \mathbf{x}_i, \mathbf{x}_j^l) d\mathbf{x}_i$. The proposal distribution $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$ for the sampling problem in (21) can be designed in a bottom-up manner, meaning that we first develop a sampling strategy and then derive the associated distribution $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$. Given r_{ij} , \mathbf{x}_j^l and the measurement model in (20), an intuitive and reasonable approach to generate \mathbf{x}_i^l is as follows:

$$\theta_{ij}^l \sim \mathcal{U}[0, 2\pi), \quad (22a)$$

$$v^l \sim f_v(v), \quad (22b)$$

$$d_{ij}^l = h^{-1}(r_{ij} - v^l), \quad (22c)$$

$$\mathbf{x}_i^l = \mathbf{x}_j^l + [d_{ij}^l \cdot \cos \theta_{ij}^l, d_{ij}^l \cdot \sin \theta_{ij}^l]^T. \quad (22d)$$

In words, the sample \mathbf{x}_i^l is obtained by moving \mathbf{x}_j^l in a random direction θ_{ij}^l by a random distance d_{ij}^l , which is generated based on the measurement model and the measurement r_{ij} . We denote the distributions of θ_{ij} , d_{ij} and \mathbf{x}_i associated with the sampling procedures in (22a), (22c) and (22d) by $q_\theta(\theta_{ij})$, $q_d(d_{ij} | r_{ij})$ and $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$, respectively. Note that the subscripts θ and d are introduced in $q_\theta(\theta_{ij})$ and $q_d(d_{ij} | r_{ij})$ to indicate the distributions of θ_{ij} and d_{ij} , respectively. However, it seems not straightforward to obtain the proposal distribution $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$.

As one of our contributions, we provide a mathematical interpretation and justification for the sampling procedure in (22), upon which, we further derive the proposal distribution $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$. The underlying idea of the sampling procedure in (22) is the transformation between a pair of random variables, from polar coordinate $[d_{ij}, \theta_{ij}]^T$ to Cartesian coordinate \mathbf{x}_i . Equivalently speaking, drawing the position sample \mathbf{x}_i^l is transformed to a problem of drawing the sample pair of distance and angle, i.e., $[d_{ij}^l, \theta_{ij}^l]^T$. As a consequence, the distributions $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$ and $q_{d,\theta}(d_{ij}, \theta_{ij} | r_{ij}) = q_d(d_{ij} | r_{ij}) \cdot q_\theta(\theta_{ij})$ are related according to

$$q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij}) = \frac{q_d(d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j^l\| | r_{ij})}{2\pi \cdot \|\mathbf{x}_i - \mathbf{x}_j^l\|}. \quad (23)$$

Thanks to (23), deriving $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$ reverts to the problem of deriving $q_d(d_{ij} | r_{ij})$, which should not be difficult for most measurement models. In our problem, where the measurement model is defined in (1), $q_d(d_{ij} | r_{ij})$ is derived as

$$q_d(d_{ij} | r_{ij}) = \frac{1}{\sqrt{2\pi} \frac{d_{ij}}{d_0} \tilde{\sigma}} \exp\left(-\frac{1}{2\tilde{\sigma}^2} \left(\log \frac{d_{ij}}{d_0} - \tilde{\mu}\right)^2\right), \quad (24a)$$

$$\tilde{\mu} = \frac{\log 10}{10\alpha^{l'}} \cdot (A - r_{ij}), \quad (24b)$$

$$\tilde{\sigma}^2 = \left(\frac{\log 10}{10\alpha^{l'}}\right)^2 \cdot \sigma^2. \quad (24c)$$

Replacing $q_d(d_{ij} | r_{ij})$ in (23) with (24) gives rise to the proposal distribution $q(\mathbf{x}_i | \mathbf{x}_j^l, r_{ij})$, which is equivalent to $q(\mathbf{x}_i | \mathbf{x}_j^l, \alpha^{l'}, r_{ij})$ in our original problem. More details about (24) are given in Appendix B.

3) *Importance Weight w_i^l* : For the auxiliary variable sample ψ^l and the position sample \mathbf{x}_i^l , which are generated from $q(\psi)$ and $q(\mathbf{x}_i | \psi^l)$, respectively, the associated importance weight w_i^l is given by

$$w_i^l \propto \frac{\prod_{j \in \Gamma_i} f(r_{ij} | \mathbf{x}_i^l, \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l})}{\sum_{j \in \Gamma_i} |\Gamma_i|^{-1} q(\mathbf{x}_i^l | \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}, r_{ij})}. \quad (25)$$

Finally, Algorithm 4 lists the steps for updating $B(\mathbf{x}_i)$ using the proposed AIS. The resulting Algorithm 1 with the beliefs updated using Algorithm 4 are named as BP-AIS or SPAWN-AIS, for that the messages are updated according to the BP or the SPAWN, respectively.

Algorithm 4 Belief Update Using AIS

- 1: **Input:** $m_{ij}^n(\mathbf{x}_i)$ for all $j \in \Gamma_i$
 - 2: **Output:** $B^n(\mathbf{x}_i) \triangleq \{\mathbf{x}_i^{l,n}\}_{l=1}^L$
 - 3: draw $\psi^l \sim q(\psi)$ as follows:
 - 4: **for** each $j \in \Gamma_i$
 - 5: draw $\psi_j^l \sim q(\psi_j)$
 - 6: **end for**
 - 7: draw $\mathbf{x}_i^l \sim q(\mathbf{x}_i | \psi^l)$ as follows:
 - 8: **for** each $j \in \Gamma_i$
 - 9: draw $\mathbf{x}_i^l \sim q(\mathbf{x}_i | \mathbf{x}_j^{\psi_j^l}, \alpha^{\psi_j^l}, r_{ij})$ using (22)
 - 10: **end for**
 - 11: calculate w_i^l using (23)-(25).
 - 12: resampling
-

VI. SOME IMPORTANT ISSUES

A. Computational Complexity

TABLE I: Complexity for updating $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$.

$m_{ij}(\alpha)$	importance weight	$\mathcal{O}(L^2)$	(7b) in BP
		$\mathcal{O}(1)$	(11a) in SPAWN
	evaluating $m_{ij}(\alpha_d^r)$	$\mathcal{O}(L \cdot R)$	
$m_{ij}(\mathbf{x}_i)$	sample α	$\mathcal{O}(\mathbb{C}_c(L, R))$	
	importance weight	$\mathcal{O}(L^2)$	(9b) in BP
		$\mathcal{O}(1)$	(11b) in SPAWN
	normalization	$\mathcal{O}(L)$	

In this subsection, the four main parts of Algorithm 1, including updating $m_{ij}(\alpha)$, $m_{ij}(\mathbf{x}_i)$, $B(\mathbf{x}_i)$ and $B(\alpha)$, will be analyzed in terms of computational complexity. To be general, we write $\mathbb{C}_c(M, N)$ to denote the complexity of drawing M samples from an N -categorical distribution. First, we consider updating $m_{ij}(\alpha)$ using Algorithm 2. Importance weights $\{w_{ij \rightarrow \alpha}^{l,n}\}_{l=1}^L$ are calculated with a complexity order of $\mathcal{O}(L^2)$ according to (7b) in the BP, but $\mathcal{O}(1)$ according to

TABLE II: Complexity for updating $B(\mathbf{x}_i)$ and $B(\alpha)$. Here, the bold fonts are used to highlight the reduction in computational complexity, where the quadratic order in the importance sampler is reduced to the linear order in the proposed AIS.

$B(\mathbf{x}_i)$	importance sampler	sampling	$\mathcal{O}(L)$
		importance weight	$\mathcal{O}(\Gamma_i \cdot L^2)$
		resampling	$\mathcal{O}(L)$
	auxiliary importance sampling	label indicator	$\mathcal{O}(\Gamma_i \cdot C_c(\frac{L}{ \Gamma_i }, L))$
		position sample	$\mathcal{O}(L)$
		importance weight	$\mathcal{O}(\Gamma_i \cdot L)$
	resampling	$\mathcal{O}(L)$	
$B(\alpha)$	evaluate $B(\alpha_d^r)$	$\mathcal{O}(\Gamma \cdot R)$	

(11a) in the SPAWN. Evaluating $m_{ij}(\alpha)$ at $\{\alpha_d^r\}_{r=1}^R$ requires operations of order $\mathcal{O}(L \cdot R)$. Second, for updating $m_{ij}(\mathbf{x}_i)$ using Algorithm 3, drawing samples $\{\alpha^l\}_{l=1}^L$ from $B(\alpha)$ needs operations of order $\mathcal{O}(C_c(L, R))$, calculating $\{w_{ij \rightarrow \mathbf{x}_i}^{l,n}\}_{l=1}^L$ has the same complexity as for $\{w_{ij \rightarrow \alpha}^{l,n}\}_{l=1}^L$, and converting $m_{ij}(\mathbf{x}_i)$ from (9a) to (10a) is done with a complexity order of $\mathcal{O}(L)$. Third, $B(\mathbf{x}_i)$ can be updated either using the importance sampler or using the proposed AIS. For the importance sampler in (15), L position samples and the corresponding importance weights are obtained with complexity orders of $\mathcal{O}(L)$ and $\mathcal{O}(|\Gamma_i| \cdot L^2)$, respectively. The subsequent resampling is conducted with a complexity order of $\mathcal{O}(L)$ [49]. For the proposed AIS in Algorithm 4, generating L label indicators has a complexity order of $\mathcal{O}(|\Gamma_i| \cdot C_c(\frac{L}{|\Gamma_i|}, L))$ approximately. Generating L position samples and calculating L importance weights according to (25) have complexity orders of $\mathcal{O}(L)$ and $\mathcal{O}(|\Gamma_i| \cdot L)$, respectively. The resampling step requires additional operations of order $\mathcal{O}(L)$. Lastly, $B(\alpha)$ is updated by simply multiplying $|\Gamma|$ real-valued numbers for R times, according to (12).

The computational complexities for updating messages and beliefs are summarized in Tables I and II, respectively. For updating $m_{ij}(\alpha)$ and $m_{ij}(\mathbf{x}_i)$, the BP is computationally substantially more intensive than the SPAWN, see Table I. Regarding updating $B(\mathbf{x}_i)$ using the importance sampler, calculating the importance weights is computationally the most intensive step, requiring operations of order $\mathcal{O}(|\Gamma_i| \cdot L^2)$. Thanks to the introduction of the auxiliary variable ψ in the proposed AIS, the quadratic complexity order is reduced to the linear order $\mathcal{O}(|\Gamma_i| \cdot L)$, see Table II.

B. Sampling From a Normalized Likelihood Function

In this subsection, we return to the problem that we have addressed in developing the AIS in Section V-B. For the general measurement model in (20), our purpose is to sample from the normalized likelihood function $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$. The sampling strategy proposed by us is essentially an importance sampler combined with random variable transformation. With the help of random variable transformation, the position samples are generated according to (22), and the associated proposal distribution $q(\mathbf{x}_i|\mathbf{x}_j^l, r_{ij})$ is derived. In the context

of this sampling problem, an importance weight, denoted by $w(\mathbf{x}_i^l)$, is assigned to the sample \mathbf{x}_i^l , as given by

$$w(\mathbf{x}_i^l) \propto \frac{f(r_{ij}|\mathbf{x}_i^l, \mathbf{x}_j^l)}{q(\mathbf{x}_i^l|\mathbf{x}_j^l, r_{ij})} = \frac{f(r_{ij}|d_{ij}^l)}{q(d_{ij}^l|r_{ij})} \cdot d_{ij}^l, \quad (26)$$

where $f(r_{ij}|d_{ij}^l)$ is $f(r_{ij}|\mathbf{x}_i^l, \mathbf{x}_j^l)$ with $\|\mathbf{x}_i^l - \mathbf{x}_j^l\|$ replaced by d_{ij}^l . This sampling strategy is related to a heuristic sampling strategy in [22]. A straightforward extension of this heuristic sampler leads to the same sample-generating mechanism, i.e., (22). Different from our sampler, these samples are deemed as following the normalized likelihood function, $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$, although they actually follow $q(\mathbf{x}_i|\mathbf{x}_j^l, r_{ij})$. A question that naturally arises is that under which condition are $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$ and $q(\mathbf{x}_i|\mathbf{x}_j^l, r_{ij})$ proportional. Referring to the relation in (26), it holds only under the condition

$$f(r_{ij}|d_{ij}) \propto q(d_{ij}|r_{ij})/d_{ij}. \quad (27)$$

Unfortunately, this condition is not fulfilled in general, and, hence, the heuristic sampler may suffer from performance loss.

Next, we will compare these two samplers in a concrete example. Consider the measurement model $r_{ij} = d_{ij} + v$ with the true distance $d_{ij} = 7.5$ and the measurement error $v \sim \mathcal{U}[-2.5, 2.5]$. For a given measurement r_{ij} and a predefined sample \mathbf{x}_j^l , the purpose is to generate a set of (weighted) samples $\{\mathbf{x}_i^l\}_{l=1}^L$ that can approximately represent the normalized likelihood function $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$. Without loss of generality, we consider that the measurement is $r_{ij} = 7.5$ and \mathbf{x}_j^l is fixed at $(0, 0)$. First, a large amount of samples are generated using (22). For our sampler, the importance weight associated with each sample is calculated using (26). Second, in order to visualize these two samplers for comparison purposes, we reconstruct the 2D plots based on the (weighted) samples, which can be done, for instance, using a (weighted) kernel density estimator (KDE) [50]. The reconstruction of our sampler and that of the heuristic sampler are illustrated in the middle and the right subplots in Fig. 3, respectively. As groundtruth, the true normalized likelihood function has been illustrated in the left subplot in Fig. 3. Several observations from Fig. 3 are listed below.

- The true normalized likelihood function (left) $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$ shows a ring-shape, implying that the measurement r_{ij} is equally probable for any position \mathbf{x}_i in the green area.
- Both samplers are able to roughly represent the ring-shape of $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$, but with different accuracy.
- The ring-shape of the heuristic sampler (right) is not equally valued. The inner-ring has higher value than the outer-ring. This implies that, using the heuristic sampling strategy, the measurement r_{ij} becomes more probable for \mathbf{x}_i located closer to \mathbf{x}_j .
- Our sampler (middle) can approximate $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j^l)$ more accurately than the heuristic sampler, since the ring-shape of our sampler is more equally valued.

Note that the difference between both samplers lies essentially in the importance weight. Our sampler computes the importance weight according to (26), for which we derive the underlying pdf $q(\mathbf{x}_i|\mathbf{x}_j^l, r_{ij})$ associated with Eq. (22) rigorously. In

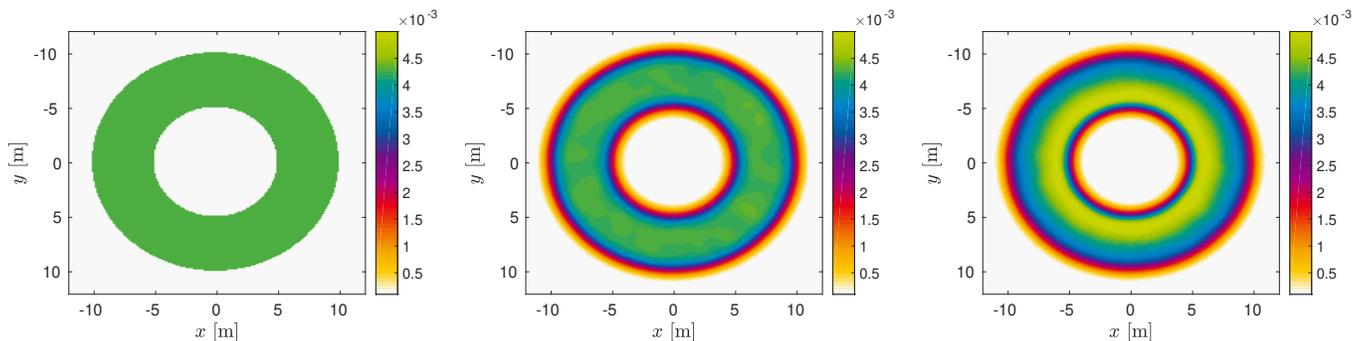


Fig. 3: A comparison between the proposed sampler (middle) and the heuristic sampler (right) versus the groundtruth (left) for sampling \mathbf{x}_i from $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j')$ for the measurement model $r_{ij} = d_{ij} + v$ with $d_{ij} = 7.5$ and $v \sim \mathcal{U}[-2.5, 2.5]$.

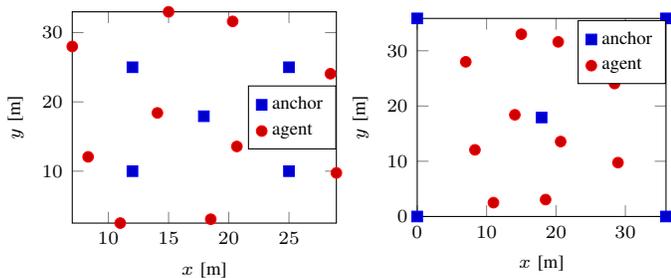


Fig. 4: Network layout: Network I (left) and Network II (right)

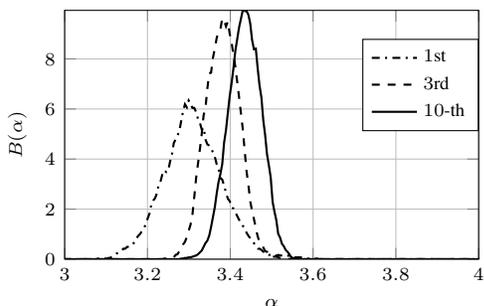


Fig. 5: Example of $B(\alpha)$ over iterations with the true $\alpha = 3.5$.

contrast, the heuristic sampling strategy ignores the mismatch between $q(\mathbf{x}_i|\mathbf{x}_j', r_{ij})$ and $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j')$, simply assigning equal weights to all samples. This example exposes the deviation of the heuristic sampler from the groundtruth. Such a deviation can be justified theoretically as well. As mentioned before, the samples of the heuristic sampler follow the desired normalized likelihood function $Z^{-1}f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j')$ under the condition in (27). However, this condition is not fulfilled here, since we have $f(r_{ij}|d_{ij}) = q_d(d_{ij}|r_{ij}) = f_v(r_{ij} - d_{ij})$ in this example. Note that this deviation becomes severe, when the likelihood function covers a broad area. On the other hand, when the likelihood function is sharp, both samplers can provide satisfactory approximation results.

VII. SIMULATION RESULTS

In this section, we comprehensively evaluate the proposed algorithms. As a competitor from the literature, we select Tomic's semidefinite programming (SDP) estimator from [17], which outperforms other existing works [15], [39]. Here, the SDP estimator is slightly adjusted so that the PLE estimate is constrained in the predefined region, in accordance with $f(\alpha)$. Note that such an adjustment can improve the original SDP estimator, since unreasonable PLE estimates are avoided. The SDP estimator terminates, either when $N_{\max} = 100$ iterations are achieved or when $|C(n) - C(n-1)|/|C(n-1)|$ is smaller than 10^{-5} , where $C(n)$ is the logarithm of the cost function in the n -th iteration. The convex optimization problem in the SDP estimator is solved using the CVX Toolbox [51] with the SeDuMi solver. For the proposed algorithms, the maximal number of iterations is set to $N_{\max} = 10$, $L = 1000$ particles are used, and $R = 100$ grid points $\{\alpha_d^r\}_{r=1}^R$ are chosen. For a fair comparison with the SDP estimator, in the proposed algorithms, a point estimate is further inferred from the marginal posterior estimate for each unknown parameter. This is done by finding the highest mode of the analytical form of $B(\mathbf{x}_i)$, which is recovered using kernel density estimation, based on the samples of $B(\mathbf{x}_i)$. Due to the fact that both the BP-IS and the BP-AIS are computationally very intensive, we will only demonstrate the performance of the SPAWN-IS and that of the SPAWN-AIS.

As shown in many experimental works, such as in [29], [32], [52], the PLE value characterizes the propagation environment, where $\alpha \in [1.5, 2]$ characterizes a LOS link, and $\alpha \in (2, 6]$ corresponds to an NLOS one. The propagation environment considered in this work is homogeneous, meaning that one common PLE α is shared by all propagation links. Accordingly, the proposed algorithms will be evaluated mainly for homogeneous environments. In Subsection VII-A, $\alpha \in \{1.5, 2, 3, 4, 5, 6\}$ are considered, corresponding to a tunnel-like scenario ($\alpha = 1.5$), a free-space scenario ($\alpha = 2$) and different NLOS scenarios ($\alpha = 3, 4, 5$ or 6). To make the simulation complete, inhomogeneous propagation environments are considered in Subsection VII-C as well.

We choose two representative networks with 10 agents and 5 anchors: Network I where some of the agents are outside

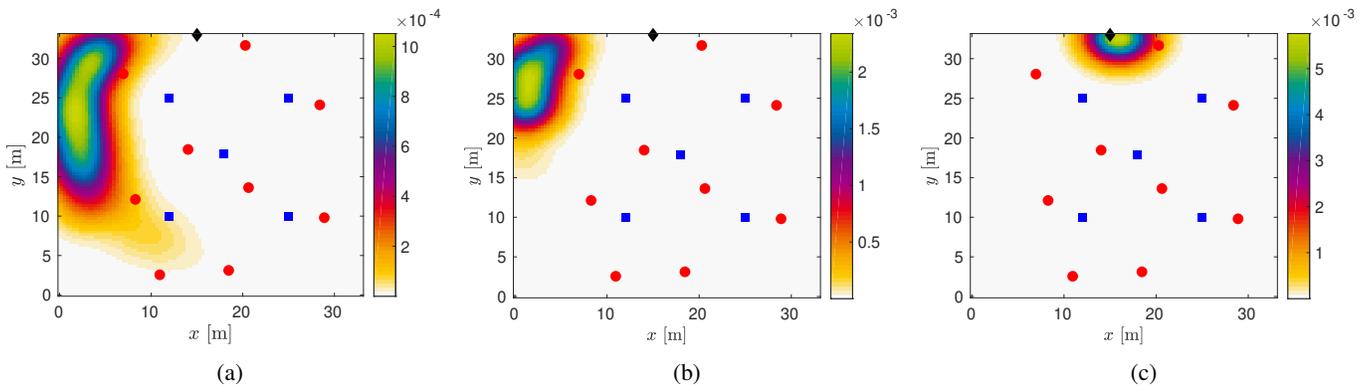


Fig. 6: Example of $B(\mathbf{x}_i)$ in the 1st (a), 3rd (b) and 10-th (c) iteration. The agent of interest locates at \blacklozenge , the anchors locate at \blacksquare , and the other agents locate at \bullet .

the convex hull of the anchors and Network II where all agents locate within the convex hull, see Fig. 4. The reference power is set to $A_i = -30$ dBm for all $i \in S$ at a reference distance of $d_0 = 1$ meter. For a fair comparison with the SDP estimator, we set the prior distribution of the PLE α as a uniform distribution, $\alpha \sim \mathcal{U}[1.5, 6]$ and that of each position as a uniform distribution in a square area that is determined by the maximum and the minimum of all nodes' positions. All simulation results are based on 100 Monte Carlo runs. The mean squared error (MSE) of the estimator $\hat{\alpha}$, the bias of $\hat{\alpha}$ and the root mean squared error (RMSE), defined in [20], are chosen as performance metrics.

A. Varying Path Loss Exponent

In this subsection, the purpose is to investigate the performance of the proposed algorithms at different PLE values in different network layouts. We set the standard deviation of the measurement error to $\sigma = 3$ and the communication range to 20 meter. As an illustrative example, we first demonstrate how the beliefs evolve with iterations and depict the kernel density estimates of $B(\alpha)$ and $B(\mathbf{x}_i)$ in Figs. 5 and 6, respectively. It is observed in Fig. 6 that over iterations $B(\mathbf{x}_i)$ becomes more concentrated and shifts towards the true position. Similarly, over iterations, the uncertainty on α reduces, and $B(\alpha)$ moves towards the true PLE $\alpha = 3.5$, see Fig. 5. It is noteworthy that the prior distributions adopted are quite coarse, for instance, a uniform distribution $\mathcal{U}[1.5, 6]$ is used for the PLE variable. Even so, the proposed algorithms can provide marginal posterior estimates that are relatively sharp and close to the true parameters.

The overall performance of different algorithms is evaluated in terms of the MSE of $\hat{\alpha}$, the bias of $\hat{\alpha}$ and the RMSE. The results are depicted in Figs. 7 and 8 for Networks I and II, respectively. For Network I, it is remarkable that, both proposed algorithms (SPAWN-IS and SPAWN-AIS) have comparable performance for both the PLE α and the position \mathbf{x}_i , $i \in S_u$, though the SPAWN-AIS has a significantly lower computational complexity than the SPAWN-IS. As compared to the SDP estimator, the proposed algorithms have improved performance for both α and \mathbf{x}_i , $i \in S_u$. For a better visualization, we depict the representative position estimates obtained

from both the SPAWN-AIS and the SDP estimator in Fig. 9. It is clear to see that the localization accuracy of the SDP estimator is quite low if part of the agents are outside the convex hull of the anchors, while the SPAWN-AIS does not suffer from this problem. We notice that this type of network topology is rarely examined in the existing literature, although its existence is very probable in practical sensor networks.

Furthermore, it is observed in Fig. 7 that the RMSE curves of the proposed algorithms first drop and then rise as the true PLE α varies. It results from two factors: the true PLE α and the estimation performance of $\hat{\alpha}$. For $\alpha \in \{1.5, 2, 3, 4, 5\}$, the proposed algorithms provide relatively accurate estimation of α . It can be approximately treated as if the PLE parameter is known. As theoretically demonstrated in [29], for RSS-based localization with known PLE, the Cramér-Rao bound is proportional to σ^2/α^2 . It means that the localization performance of RSS-based algorithms generally improves when the ratio σ/α decreases (when α increases for a fixed σ). This explains why the RMSE curves first drop. On the other hand, when the true underlying α is 6, $\hat{\alpha}$ tends to underestimate α . The influence of an incorrect PLE (bias of $\hat{\alpha}$) on RSS-based localization has been studied in [34]. As shown in [34], the localization error becomes considerably large, when $\hat{\alpha}$ underestimates α . This explains why the curves rise at $\alpha = 6$.

For Network II, again, the MSE curve of $\hat{\alpha}$ of the proposed algorithms is under that of the SDP estimator, see Fig. 8, meaning that the proposed algorithms have quite stable estimation performance for the PLE α . However, for this network, the localization accuracy of the proposed algorithms is comparable to or slightly lower than that of the SDP estimator. This localization performance degradation in the proposed algorithms results from a biased estimation of α , which can be seen in the plot on the bottom left in Fig. 8. The possible reason lies in that there could be certain performance loss when we infer the unknown parameter from its marginal posterior, instead of jointly inferring all unknown parameters from the joint posterior. Nevertheless, this problem will be alleviated either when the communication range increases or when the measurement noise decreases, as will be demonstrated in the following simulations.

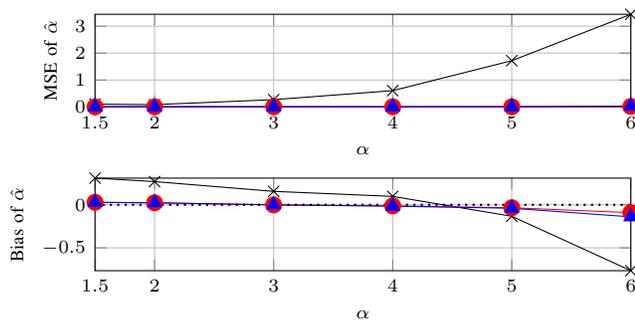


Fig. 7: Network I: The MSE of $\hat{\alpha}$ (top left), the bias of $\hat{\alpha}$ (bottom left) and the RMSE (right) versus the true PLE α . Here, the standard deviation of the measurement error is $\sigma = 3$, and the communication range is 20 meter.

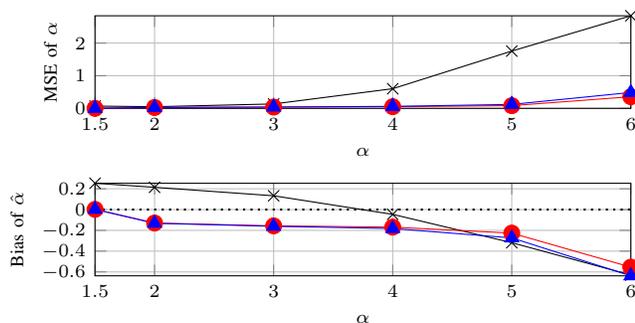


Fig. 8: Network II: The MSE of $\hat{\alpha}$ (top left), the bias of $\hat{\alpha}$ (bottom left) and the RMSE (right) versus the true PLE α . Here, the standard deviation of the measurement error is $\sigma = 3$, and the communication range is 20 meter.

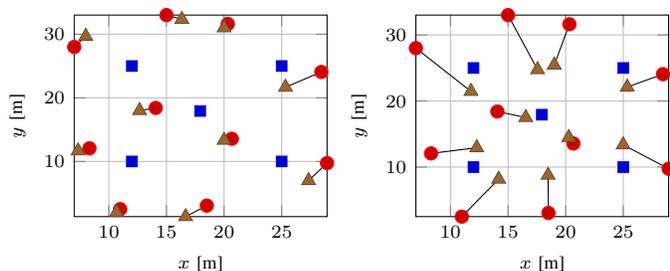
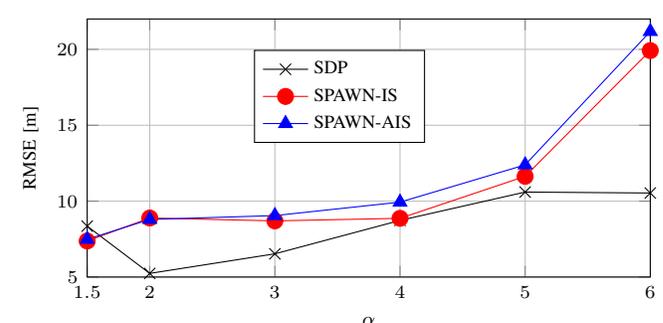
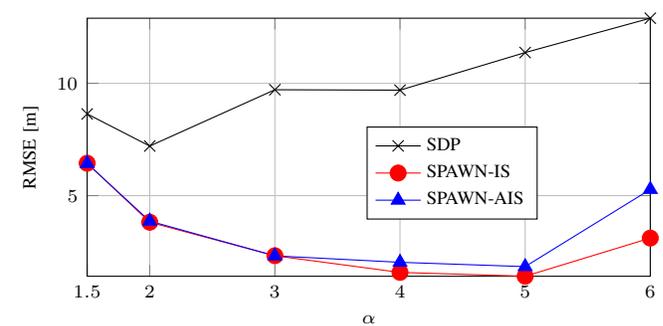


Fig. 9: Position estimates obtained by the SPAWN-AIS (left) and by the SDP (right) with anchors \blacksquare , agents \bullet , estimated agents \blacktriangle .

B. Varying Communication Range and Standard Deviation

The purpose of this subsection is to assess the performance of the proposed algorithms at varying communication range and varying standard deviation of the measurement error. It has been shown that for Network I the proposed algorithms have quite satisfying performance for both the positions and the PLE. Hence, in the following simulations, we will only focus on Network II. For the simulation with varying communication range, the true PLE and the standard deviation are set to $\alpha = 3$ and $\sigma = 3$, respectively, and for the other simulation, the true PLE is set to $\alpha = 3$, and the communication range is set to 25 meter. The results are depicted in Figs. 10 and 11 for the cases of varying communication range and varying standard deviation, respectively.

From the figures we can see that for the proposed algorithms, all three error-curves drop substantially and eventually attain relatively small values, as the communication range



increases or the standard deviation decreases. In contrast, no obvious improvement is seen for the SDP estimator. This result is expected and can be explained as follows. In the proposed Bayesian algorithms, the marginal posterior of each unknown parameter is inferred. When more information is collected, for instance through increasing communication range (network connectivity) or through decreasing measurement error, the marginal posterior can reflect the unknown parameter more accurately. On the other hand, the SDP estimator suffers from certain performance loss due to the relaxation procedure, and this performance loss may be so dominating that the increase in the information cannot improve the estimation accuracy any more. This result highlights that the proposed algorithms can benefit from the increase in the information to a large extent. Lastly, we stress that although the SPAWN-AIS has a significant reduction on computational cost, it achieves similar estimation performance as the SPAWN-IS.

C. Inhomogeneous Propagation Environments

Here, we consider the case when part of the propagation links are LOS links and the others are NLOS links. The probability that one measurement is in the NLOS condition is denoted by p_{NLOS} . To represent different scenarios, it varies from 0.1 to 0.9 with a stepsize of 0.2. A LOS measurement is generated using (1) with $\alpha = 2$; while an NLOS one is generated with $\alpha = 4$. The other parameters stay the same as in Subsection VII-A. The localization performance of all algorithms is depicted in Fig. 12 for both networks. Certain performance degradation has been observed for the

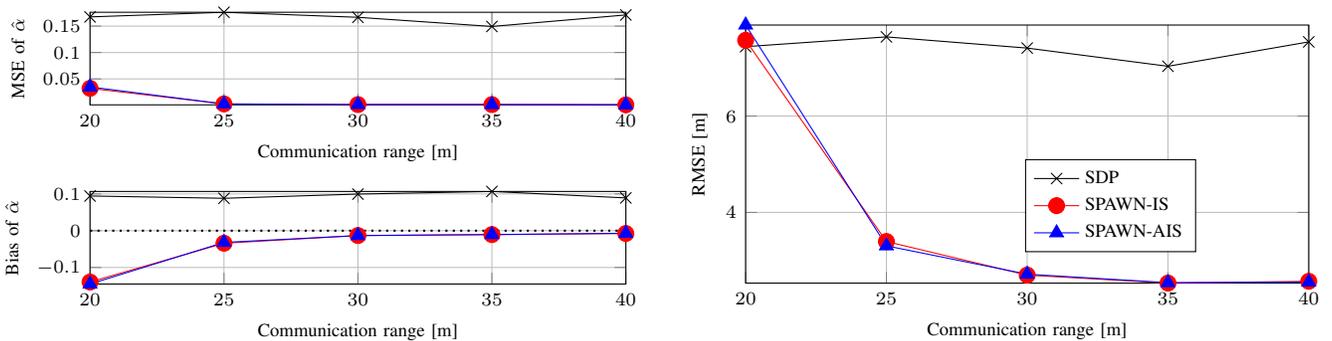


Fig. 10: Network II: The MSE of $\hat{\alpha}$ (top left), the bias of $\hat{\alpha}$ (bottom left) and the RMSE (right) versus the communication range. Here, the standard deviation of the measurement error is $\sigma = 3$, and the true underlying PLE is $\alpha = 3$.

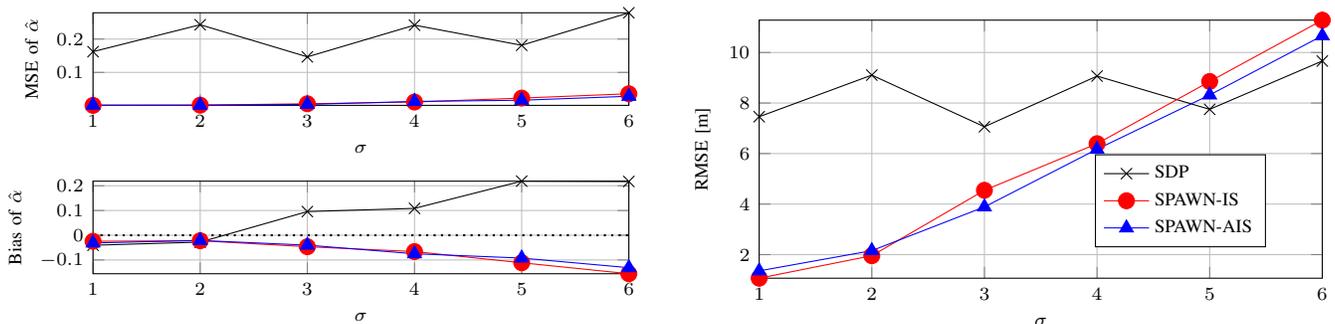


Fig. 11: Network II: The MSE of $\hat{\alpha}$ (top left), the bias of $\hat{\alpha}$ (bottom left) and the RMSE (right) versus the standard deviation of the measurement error. Here, the true underlying PLE is $\alpha = 3$, and the communication range is 25 meter.

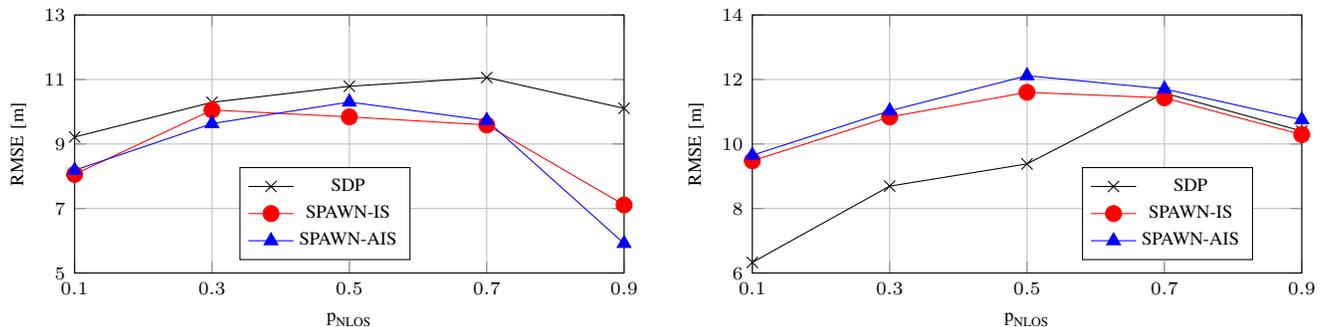


Fig. 12: Performance obtained in the inhomogeneous environments for Network I (left) and Network II (right). Part of the propagation links (LOS) have $\alpha = 2$ and the others (NLOS) have $\alpha = 4$. The probability that one link is in the NLOS condition, denoted as p_{NLOS} , varies from 0.1 to 0.9 with a stepsize of 0.2. Here, the standard deviation of the measurement is $\sigma = 3$, and the communication range is 25 meter.

proposed algorithms (SPAWN-IS and SPAWN-AIS), and for the competing algorithm as well. Although the proposed algorithms have lower localization accuracy for network II, they consistently outperform the SDP estimator for network I. It can be seen from Fig. 12 that almost all error-curves follow the ascending-descending trend, for instance, the RMSE curve of the SPAWN-IS algorithm in the left plot starts from 8 [m] at $p_{\text{NLOS}} = 0.1$, approaching 10 [m] at $p_{\text{NLOS}} = 0.5$, ends at 7 [m] at $p_{\text{NLOS}} = 0.9$. This observation implies that as the propagation environment deviates more from a homogeneous one, the performance degradation becomes more prominent. However, such performance degradation is expected, because none of these algorithms are developed for the inhomogeneous environments.

Although developing RSS-based cooperative localization algorithm for inhomogeneous environments is beyond the scope of this work, a discussion on this issue is important. If no prior experimental campaign is conducted, RSS-based cooperative localization for inhomogeneous environments is a challenging problem and has been rarely studied. For the purpose of promoting the development in this direction, several existing noncooperative RSS-based algorithms are shortly reviewed. In [53], the unknown and unequal PLEs are estimated using measures of compatibility of the distance estimates. In [54], a generalized two-region channel model is used and the PLE in the NLOS region is adapted online while positioning. Similarly, a mode-dependent propagation model is proposed in [37] and all unknown parameters are jointly estimated using

the expectation-maximization criterion. Moreover, the recent work [55] has proposed a piecewise convex approximation aided localization dealing with the nonlinear and nonconvex problem, resulting from the unknown and unequal PLEs.

If it is affordable to conduct an experimental campaign to build a database, the fingerprinting method [56] and its variations can generally provide quite satisfying localization performance. However, the major drawbacks of the fingerprinting approaches lie in the requirement of a database and the inability to adapt to a new environment. Furthermore, besides the database, if more advanced measurements can be collected, such as the waveform of UWB signals, there are various potential solutions for cooperative localization in inhomogeneous environments. One potential solution is the paradigm based on the concept of soft-range-information (SRI) [57], [58]. In contrast to the conventional techniques that rely on the most likely distance estimate, SRI enables soft-decision localization by capturing the odds of all possible distances. Based on the training data collected during the experimental campaign, a generative model is estimated via machine learning and then stored, and the SRI is estimated using the stored generative model. In addition, the NLOS identification and mitigation techniques might potentially improve the localization performance in inhomogeneous environments. Much attention has been paid for TOA measurements (especially, UWB signal), e.g., in [59]–[61], but there are only limited works on RSS, which is extracted from signals with much narrower bandwidths, such as Zigbee signals. Relying on the training data collected in an experimental campaign, NLOS identification using RSS data has been studied in the recent work [62].

VIII. CONCLUSION

In this paper, we addressed the problem of RSS-based cooperative localization with unknown PLE in a homogeneous environment using a Bayesian framework. To infer the marginal posterior of each unknown parameter, we have developed a series of message passing algorithms. The proposed algorithms provide a unified strategy for estimating both the positions and the PLE parameter and, therefore, handle the problem from a more theoretical perspective, as compared to the heuristic alternating strategy in the existing literature. Our comprehensive simulation studies prove the validity of our procedure with four major results. (1) As compared to the competitor, the proposed algorithms achieve comparable or better localization performance, depending on the network layout; (2) The proposed algorithms can benefit from the increase in the information significantly and tend to outperform the existing one in dense networks and low-to-medium noise scenarios; (3) Concerning the PLE parameter, the proposed algorithms tend to underestimate it, incurring deterioration of localization accuracy. Nevertheless, the proposed algorithms consistently achieve a smaller MSE than the competitor; (4) Among the proposed algorithms, the SPAWN-AIS achieves comparable performance, but at the lowest computational cost. Many research challenges need to be overcome in our future work, including reducing the bias in the PLE and extending this work to an inhomogeneous environment.

APPENDIX A DERIVATION OF (10c)

We consider the calculation of the integral in (10c). For notational simplicity, we ignore the superscript $l, n-1$ during this derivation. The integral is calculated as follows:

$$\begin{aligned} Z_{ij} &= \int f(r_{ij}|\mathbf{x}_i, \mathbf{x}_j, \alpha) \, d\mathbf{x}_i \\ &= \int f_{\mathcal{N}}\left(r_{ij} - A_i + 10\alpha \log_{10} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{d_0}\right) \, d\mathbf{x}_i \\ &\stackrel{\textcircled{1}}{=} \int f_{\mathcal{N}}\left(r_{ij} - A_i + 10\alpha \log_{10} \frac{\|\mathbf{x}_{ij}\|}{d_0}\right) \, d\mathbf{x}_{ij} \\ &\stackrel{\textcircled{2}}{=} \int_0^{2\pi} \int_0^{\infty} f_{\mathcal{N}}\left(r_{ij} - A_i + 10\alpha \log_{10} \frac{d_{ij}}{d_0}\right) \cdot d_{ij} \, dd_{ij} \, d\theta_{ij} \\ &= 2\pi \frac{\log 10}{10\alpha} \mathbb{E}_{d_{ij} \sim \log \mathcal{N}(\mu_d, \sigma_d^2)} [d_{ij}^2] \\ &= 2\pi \frac{\log 10}{10\alpha} \cdot \exp(2\sigma_d^2 + 2\mu_d) \end{aligned}$$

with

$$\mu_d = \frac{\log 10}{10\alpha} (A_i - r_{ij}) + \log d_0, \quad \sigma_d^2 = \sigma^2 \frac{(\log 10)^2}{(10\alpha)^2}.$$

Here, $f_{\mathcal{N}}(\cdot)$ stands for the pdf of the Gaussian distribution $\mathcal{N}(0, \sigma^2)$, $\textcircled{1}$ stands for $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$, and from $\textcircled{1}$ to $\textcircled{2}$ is achieved by transforming the Cartesian coordinate \mathbf{x}_{ij} to the polar coordinate $[d_{ij}, \theta_{ij}]^T$.

APPENDIX B DERIVATION OF (24a)

For the measurement model in (1), the distance sample d_{ij}^l generated according to (22c)-(22b) fulfills the relation

$$\log \frac{d_{ij}^l}{d_0} = \underbrace{\frac{\log 10}{10\alpha^{l'}} \cdot (A_i - r_{ij}) + \frac{\log 10}{10\alpha^{l'}} \cdot v^l}_{\tilde{v}}.$$

Given α , A_i and r_{ij} , the variable \tilde{v} is Gaussian distributed, namely, $\tilde{v} \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$ with

$$\tilde{\mu} = \frac{\log 10}{10\alpha^{l'}} \cdot (A_i - r_{ij}), \quad \tilde{\sigma}^2 = \left(\frac{\log 10}{10\alpha^{l'}}\right)^2 \cdot \sigma^2.$$

It follows that d_{ij}/d_0 is log-normal distributed, namely,

$$d_{ij}/d_0 \sim \text{LogN}(\tilde{\mu}, \tilde{\sigma}^2).$$

Furthermore, it is given that the pdf of a log-normal-distributed random variable $a \sim \text{LogN}(\mu_a, \sigma_a^2)$ is in the form of

$$f(a) = \frac{1}{\sqrt{2\pi}a\sigma_a} \exp\left(-\frac{(\log a - \mu_a)^2}{2\sigma_a^2}\right). \quad (29)$$

Finally, substituting a , μ_a and σ_a in (29) with d_{ij}/d_0 , $\tilde{\mu}$ and $\tilde{\sigma}^2$, respectively, concludes the derivation.

ACKNOWLEDGMENT

The authors would like to thank Mr. Slavisa Tomic for providing the MATLAB codes of the comparative algorithm. The work of Feng Yin was supported mainly by the NSFC under Grant 61701426, and in part by Guangdong Province Grant 2017ZT07X152, and in part by the Shenzhen Fundamental Research Fund under Grant ZDSYS201707251409055.

REFERENCES

- [1] J. Ko, T. Gao, R. Rothman, and A. Terzis, "Wireless Sensing Systems in Clinical Environments: Improving the Efficiency of the Patient Monitoring Process," *IEEE Eng. Med. Biol. Mag.*, vol. 29, no. 2, pp. 103–109, March 2010.
- [2] COST Action CA15104, IRACON, T. Pedersen, and B. Fleury, "Whitepaper on New Localization Methods for 5G Wireless Systems and the Internet-of-Things," 2018.
- [3] R. Karlsson and F. Gustafsson, "The Future of Automotive Localization Algorithms: Available, reliable, and scalable localization: Anywhere and anytime," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 60–69, March 2017.
- [4] F. Zabini and A. Conti, "Inhomogeneous Poisson Sampling of Finite-Energy Signals With Uncertainties in \mathbb{R}^d ," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4679–4694, Sep. 2016.
- [5] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Tasilila, and R. Curtmola, "Fostering Participation in Smart Cities: A Geosocial Crowdsensing Platform," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, June 2013.
- [6] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug 2002.
- [7] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating The Nodes: Cooperative Localization in Wireless Sensor Networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, 2005.
- [8] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative Localization in Wireless Networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, 2009.
- [9] S. H. Li, W. Ni, C. K. Sung, and M. Hedley, "Recent Advances on Cooperative Wireless Localization and Their Application in Inhomogeneous Propagation Environments," *Computer Networks*, vol. 142, pp. 253 – 271, 2018.
- [10] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative Sensor Network Localization: Algorithms and Practical Issues," *Proc. IEEE*, vol. 106, no. 6, pp. 1089–1114, 2018.
- [11] Y. Shen, H. Wymeersch, and M. Z. Win, "Fundamental Limits of Wideband Localization—Part II: Cooperative Networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4981–5000, 2010.
- [12] J. Schloemann and R. M. Buehrer, "On the Value of Collaboration in Location Estimation," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3585–3596, 2016.
- [13] P. Tseng, "Second-Order Cone Programming Relaxation of Sensor Network Localization," *SIAM J. Optimization*, vol. 18, no. 1, pp. 156–185, 2007.
- [14] R. W. Ouyang, A. K.-S. Wong, and C.-T. Lea, "Received Signal Strength-Based Wireless Localization via Semidefinite Programming: Noncooperative and Cooperative Schemes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1307–1318, 2010.
- [15] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, "Cooperative Received Signal Strength-Based Sensor Localization With Unknown Transmit Powers," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, 2013.
- [16] A. Simonetto and G. Leus, "Distributed Maximum Likelihood Sensor Network Localization," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, 2014.
- [17] S. Tomic, M. Beko, and R. Dinis, "RSS-Based Localization in Wireless Sensor Networks Using Convex Relaxation: Noncooperative and Cooperative Schemes," *IEEE Trans. Veh. Technology*, vol. 64, no. 5, pp. 2037–2050, 2015.
- [18] J. Costa, N. Patwari, and A. Hero, "Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks," *ACM Trans. Sensor Networks*, vol. 2, no. 1, pp. 39–64, 2006.
- [19] X. Li, "Collaborative Localization With Received-Signal Strength in Wireless Sensor Networks," *IEEE Trans. Veh. Technology*, vol. 56, no. 6, pp. 3807–3817, 2007.
- [20] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, "Cooperative Localization in WSNs Using Gaussian Mixture Modeling: Distributed ECM Algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1448–1463, 2015.
- [21] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric Belief Propagation," in *IEEE Conf. Computer Vision Pattern Recognition*, vol. 1, 2003.
- [22] A. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric Belief Propagation for Self-Localization of Sensor Networks," in *IEEE J. Sel. Areas Commun.*, 2005.
- [23] J. Lien, U. J. Ferner, W. Srichavengsup, H. Wymeersch, and M. Z. Win, "A Comparison of Parametric and Sample-Based Message Representation in Cooperative Localization," *Hindawi Publishing Corporation*, 2012.
- [24] S. Li, M. Hedley, and I. B. Collings, "New Efficient Indoor Cooperative Localization Algorithm With Empirical Ranging Error Model," *IEEE J. Selected Areas Commun.*, vol. 33, no. 7, pp. 1407–1417, 2015.
- [25] D. Jin, F. Yin, C. Fritsche, A. M. Zoubir, and F. Gustafsson, "Efficient Cooperative Localization Algorithm in LOS/NLOS Environments," in *European Signal Process. Conf. (EUSIPCO)*, 2015, pp. 185–189.
- [26] F. Scheidt, D. Jin, M. Muma, and A. M. Zoubir, "Fast and Accurate Cooperative Localization in Wireless Sensor Networks," in *European Signal Process. Conf. (EUSIPCO)*, 2016.
- [27] F. Viani, P. Rocca, G. Oliveri, D. Trinchero, and A. Massa, "Localization, Tracking, And Imaging of Targets in Wireless Sensor Networks: An Invited Review," *Radio Sci.*, vol. 46, no. 5, pp. 1–12, 2011.
- [28] WG802.15, "IEEE standards for low-rate wireless networks," 2015.
- [29] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, 2003.
- [30] X. Li, "RSS-Based Location Estimation with Unknown Pathloss Model," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3626–3633, 2006.
- [31] G. Q. Mao, B. D. O. Anderson, and B. Fidan, "Path Loss Exponent Estimation for Wireless Sensor Network Localization," *Computer Networks*, vol. 51, no. 10, pp. 2467 – 2483, 2007.
- [32] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.
- [33] N. Salman, M. Ghogho, and A. H. Kemp, "On the Joint Estimation of the RSS-Based Location and Path-loss Exponent," *IEEE Wireless Commun. Lett.*, vol. 1, no. 1, pp. 34–37, 2012.
- [34] N. Salman, A. H. Kemp, and M. Ghogho, "Low Complexity Joint Estimation of Location and Path-Loss Exponent," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 364–367, 2012.
- [35] Y. T. Chan, B. H. Lee, R. Inkol, and F. Chan, "Received Signal Strength Localization with An Unknown Path Loss Exponent," in *24th Canadian Conf. Electr. Computer Eng. (CCECE)*, 2011, pp. 000456–000459.
- [36] F. Gustafsson, F. Gunnarsson, and D. Lindgren, "Sensor models and localization algorithms for sensor networks based on received signal strength," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 16, Jan 2012.
- [37] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Received Signal Strength-Based Joint Parameter Estimation Algorithm for Robust Geolocation in LOS/NLOS Environments," in *IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2013, pp. 6471–6475.
- [38] H. C. So and L. Lin, "Linear Least Squares Approach for Accurate Received Signal Strength Based Source Localization," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 4035–4040, 2011.
- [39] G. Wang, H. Chen, Y. Li, and M. Jin, "On Received-Signal-Strength Based Localization with Unknown Transmit Power and Path Loss Exponent," *IEEE Wireless Commun. Lett.*, vol. 1, no. 5, pp. 536–539, 2012.
- [40] F. Bandiera, A. Coluccia, and G. Ricci, "A Cognitive Algorithm for Received Signal Strength Based Localization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1726–1736, April 2015.
- [41] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, R. J. Brachman, Ed. Morgan Kaufmann Publishers, 1988.
- [42] A. Ihler and D. McAllester, "Particle Belief Propagation," in *Int. Conf. AI Statistics*, 2009.
- [43] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric Belief Propagation," *Commun. ACM*, vol. 53, no. 10, pp. 95–103, 2010.
- [44] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [45] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient Multiscale Sampling From Products of Gaussian Mixtures," in *Advances Neural Inf. Proc. Systems*. MIT Press, 2004, pp. 1–8.
- [46] D. Rudoy and P. J. Wolfe, "Multi-Scale MCMC Methods for Sampling from Products of Gaussian Mixtures," in *IEEE Int. Conf. Acoustics, Speech Signal Proc. (ICASSP)*, vol. 3, 2007, pp. 1201–1204.
- [47] M. Briers, A. Doucet, and S. S. Singh, "Sequential Auxiliary Particle Belief Propagation," in *Int. Conf. Inf. Fusion*, vol. 1, 2005.
- [48] D. Jin, F. Yin, C. Fritsche, A. M. Zoubir, and F. Gustafsson, "Cooperative Localization Based on Severely Quantized RSS Measurements in Wireless Sensor Network," in *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, 2016.

- [49] J. D. Hol, T. B. Schön, and F. Gustafsson, "On Resampling Algorithms for Particle Filters," in *IEEE Nonlinear Statistical Signal Process. Workshop*, 2006, pp. 79–82.
- [50] B. W. Silverman, *Density Estimation: for Statistical and Data Analysis*, 1st ed. London, UK: Chapman and Hall, 1986.
- [51] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," <http://cvxr.com/cvx>, 2017.
- [52] A. Conti, D. Dardari, M. Guerra, L. Mucchi, and M. Z. Win, "Experimental Characterization of Diversity Navigation," *IEEE Syst. J.*, vol. 8, no. 1, pp. 115–124, March 2014.
- [53] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 821–831, Oct 2009.
- [54] M. F. Mansour, "Channel learning in indoor localization," in *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, March 2016, pp. 3886–3890.
- [55] B. Liu, X. Zhu, Y. Jiang, Z. Wei, and Y. Huang, "UAV and Piecewise Convex Approximation Assisted Localization With Unknown Path Loss Exponents," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 12396–12400, Dec 2019.
- [56] P. Bahl and V. N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," in *Conf. Computer Commun.*, vol. 2, March 2000, pp. 775–784 vol.2.
- [57] S. Mazuelas, A. Conti, J. C. Allen, and M. Z. Win, "Soft Range Information for Network Localization," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3155–3168, June 2018.
- [58] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft Information for Localization-of-Things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov 2019.
- [59] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network Experimentation for Cooperative Localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, February 2012.
- [60] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS Identification and Mitigation for Localization Based on UWB Experimental Data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.
- [61] D. Dardari, A. Conti, J. Lien, and M. Z. Win, "The Effect of Cooperation on Localization Systems Using UWB Experimental Data," *EURASIP J. Advances Signal Process.*, vol. 2008, no. 1, Feb 2008.
- [62] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-Line-of-Sight Identification and Mitigation Using Received Signal Strength," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1689–1702, March 2015.



Di Jin received the B.Sc. degree in information and communication engineering from Zhejiang University, Hangzhou, China in 2011, and the M.Sc. degree in electrical engineering and information technology from Technische Universität Darmstadt, Darmstadt, Germany, in 2014. She is currently working towards the Ph.D. degree in the Signal Processing Group at Technische Universität Darmstadt. Her research interests include localization and tracking, distributed and cooperative inference in wireless networks.



Feng Yin (M'14) received the B.Sc. degree from Shanghai Jiao Tong University, China, in 2008, and the M.Sc. and Ph.D. degrees from Technische Universität Darmstadt, Germany, in 2011 and 2014, respectively. He is currently an assistant professor in the School of Science and Engineering at the Chinese University of Hong Kong, Shenzhen, China. From 2014 to 2016, he was with Ericsson Research, Linköping, Sweden, working on the European Union FP7 Marie Curie Training Programme on Tracking in Complex Sensor Systems (TRAX). Since 2016, he has been with The Chinese University of Hong Kong, Shenzhen and also affiliated with the Shenzhen Research Institute of Big Data (SRIBD). His research interests include statistical signal processing, Bayesian deep learning, and sensory data fusion. He was a recipient of the Chinese Government Award for Outstanding Self-Financed Students Abroad in 2013. He received the Marie Curie Scholarship from the European Union in 2014. He is currently serving as the handling editor of the Elsevier Signal Processing and Elsevier Digital Signal Processing.



Carsten Fritsche (M'10) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering and information technology from Technische Universität Darmstadt, Darmstadt, Germany, in 2005 and 2011, respectively. In 2011, he joined the Division of Automatic Control, Linköping University, Linköping, Sweden, as a Postdoctoral Research Fellow. In 2012, he held various positions in industry and is now with IAV GmbH, Berlin, Germany, where he is working on the next generation of advanced driver assistance systems. Since summer 2014, he has been additionally with the Division of Automatic Control, Linköping University, where he is currently an Adjunct Associate Professor. His main research interests include statistical signal processing, machine learning, and sensor fusion with applications to wireless positioning, satellite navigation, and driver assistance systems.



Fredrik Gustafsson (F'11) received the M.Sc. degree in electrical engineering and the Ph.D. degree in automatic control from Linköping University, Linköping, Sweden, in 1988 and 1992, respectively. Since 2005, he has been a Professor in sensor informatics with the Department of Electrical Engineering, Linköping University. His research interests include stochastic signal processing, adaptive filtering, and change detection, with applications to communication, vehicular, airborne, and audio systems. He is a cofounder of the companies NIRA Dynamics (automotive safety systems), Softube (audio effects) and SenionLab (indoor positioning systems). He was an Associate Editor for the IEEE Transactions of Signal Processing (2000–2006), IEEE Transactions on Aerospace and Electronic Systems (2010–2012), and EURASIP Journal on Applied Signal Processing (2007–2012). He was the recipient of Arberg prize by the Royal Swedish Academy of Science (KVA) 2004 and an elected member of the Royal Academy of Engineering Sciences (IVA) 2007. He was the recipient of the Harry Rowe Mimno Award 2011 for the tutorial "Particle Filter Theory and Practice with Positioning Applications", which was published in the AESS Magazine in July 2010, and was the coauthor of "Smoothed state estimates under abrupt changes using sum-of-norms regularization" that received the Automatica Paper Prize in 2014.



Abdelhak M. Zoubir (S'87–M'92–SM'97–F'08) received the Dr.-Ing. degree from Ruhr-Universität Bochum, Bochum, Germany, in 1992. He was with the Queensland University of Technology, Australia, from 1992–1998 where he was an Associate Professor. In 1999, he joined the Curtin University of Technology, Australia, as a Professor of Telecommunications and was Interim Head of the School of Electrical and Computer Engineering from 2001 to 2003. In 2003, he moved to Technische Universität Darmstadt, Germany, as a Professor of Signal Processing and Head of the Signal Processing Group. His research interest lies in statistical methods for signal processing with emphasis on bootstrap techniques, robust detection and estimation and array processing applied to telecommunications, radar, sonar, car engine monitoring and safety, and biomedicine. He has authored or coauthored more than 400 journal and conference papers on these areas. Professor Zoubir was the General or Technical Chair of numerous international workshops and conferences, notably the Technical Chair of the IEEE International Conference on Acoustics, Speech, and Signal Processing held in 2014 in Florence (Italy). He was also on several Editorial Boards and was a Guest Editor on numerous occasions, most notably he was the Editor-in-Chief of the IEEE Signal Processing Magazine from 2012 to 2014. He served on various IEEE SPS Technical Committees, notably he was a Chair (2010–2011) of the IEEE SPS Technical Committee Signal Processing Theory and Methods. He served on the Board of Governors of the IEEE SPS as an elected Member-at-Large (2015–2017), and was an Elected Member of the Board of Directors of the European Association of Signal Processing (EURASIP) from 2008 to 2016, and its President from 2017 to 2018. He is an IEEE Distinguished Lecturer (Class 2010–2011). He was the recipient of several awards, most recently the 2018 Leo L. Beranek Meritorious Service Award and the 2017 IEEE Signal Processing Magazine Best Paper Award.