Received May 30, 2018, accepted June 28, 2018, date of publication September 4, 2019, date of current version September 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2861387

Cooperative Localization in Mobile Wireless Networks With Asynchronous Measurements and Communications

YIRUI CONG[1], (Member, IEEE), XIANGYUN ZHOU[2], (Senior Member, IEEE), BINGPENG ZHOU[3], (Member, IEEE), AND VINCENT K. N. LAU[3], (Fellow, IEEE)
[1]College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
[2]Research School of Electrical, Energy and Materials Engineering, Australian National University, Canberra, ACT 0200, Australia
[3]Department of ECE, Hong Kong University of Science and Technology, Hong Kong

Corresponding author: Yirui Cong (congyirui11@nudt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61801494 and Grant 61876187.

ABSTRACT This paper studies the cooperative localization problem for mobile nodes. Different from the previous work which highly relies on the synchronized time-slotted systems, the cooperative localization framework we establish does not need any synchronization for the communication links and measurement processes in the entire wireless network. More specifically, the mobility of nodes is modeled by the stochastic differential equation, which allows the modeling of each node utilizing the asynchronously received messages and measurements in the continuous-time domain. To solve the cooperative localization problem, first we propose the centralized localization algorithm based on the global information. Then, we rigorously prove under what condition a localization estimation with partial information has a small performance gap from the one with global information. Finally, by applying this result at each node, the distributed prior-cut algorithm is designed to solve this asynchronous localization problem. Two cooperative localization examples are presented to corroborate the effectiveness of the distributed prior-cut algorithm in dealing with asynchronous communications and measurements.

INDEX TERMS Cooperative localization, wireless networks, asynchronous, stochastic differential equation, distributed prior-cut algorithm, continuous-discrete filter.

I. INTRODUCTION

A. MOTIVATION

Cooperative localization in mobile wireless networks has attracted significant attention in recent years, and is becoming increasingly important for various in location-aware applications [1], [2]. To estimate its location, each wireless device (node) works cooperatively with other nodes in the wireless network. Generally speaking, there are three key elements in any cooperative localization algorithm:

1) receiving location-related information from other nodes (information collecting);
2) estimating location based on the received information so far (information inferring);
3) transmitting new location-related information to other nodes (information sharing).

The information collecting-inferring-sharing framework is very helpful to understand the existing work and develop new techniques.

The associate editor coordinating the review of this article and approving it for publication was Ligang Wu.

For mobile nodes, the cooperative localizations are usually based on the sequential Bayesian filtering technique in time-slotted systems. That means all the information in the wireless network must be collected, inferred, and shared in a synchronized manner according to the time-slotted system. Without synchronization, the existing cooperative localization algorithms for mobile nodes cannot work properly. In the very recent years, the importance of the synchronization in mobile wireless networks has been realized, and clock synchronization problems were considered in [3] and [4].

In practical scenarios, however, it is difficult to synchronize all the clocks, communication links, and sensor measurements, especially in the large-scale wireless networks. This fact motivates us to rethink the sequential Bayesian filtering framework: whether cooperative localization can be done in an asynchronous way without relying on the synchronized measurements and communications? More specifically, how to model and solve the cooperative localization problem for mobile nodes with asynchronous measurements and communications? In this paper, we aim to provide a solution to this problem.

B. RELATED WORK

Since this work represents the first step towards solving the asynchronous cooperative localization problem for mobile nodes, there is no closely related work in the sense of asynchronous communications and measurements. We review the Belief-Propagation (BP) based cooperative localization and the continuous-discrete Bayesian filter, because they inspire us the most in this work.

BP localization is based on the factor graph [5] in the Bayesian filtering framework [6]. It marginalizes the joint posterior into approximated marginal posteriors (i.e. beliefs) by iteratively exchanging local information (i.e. scheduled message passing) among a given cluster of nodes, where each marginal posterior refers to the posterior of the location-related state in the corresponding node. Employing the BP method in localization problems can date back to the study on static nodes [7], which was then generalized to the famous Sum-Product Algorithm over a Wireless Network (SPAWN) method for mobile nodes by the celebrated work in [1]. Note that for SPAWN the information is collected, inferred, and shared multiple times (i.e. multiple iterations) in each time slot, which can have complexity and overhead issues in communications. To solve this problem, a series of improvements were studied to reduce the computational complexity and communication (iteration) times [8]–[12]. In [8], a Gaussian mixture model replaced the particles to describe the beliefs, and the communication scheme was modified to only include beliefs (rather than both beliefs and messages) where Kullback Leibler divergence was employed to censor if a belief is good enough to transmit. In [9], the sigma point BP was proposed to reduce the low communication overhead. Based on the sigma point BP, [10] designed a greedy algorithm for further reducing the communication overhead. In [11] and [12], the BP method was combined with the mean-field approximation to reduce the message size in transmission.

In terms of the continuous-discrete Bayesian filter, it is considered in the continuous-time domain, and its measurements can be observed at any discrete time instants without time-slotted constraints. If a cooperative localization problem can be modeled in this continuous-discrete framework, then the asynchronous communications and measurements will automatically fit in this model.

Compared to the discrete Bayesian filter, the continuous-discrete Bayesian filter has not drawn much attention since R. E. Kalman proposed his famous Kalman filter [13]. The continuous-discrete Kalman filter as well as the extended Kalman filter were presented in the Bayesian framework in [14]. The unscented Kalman, cubature Kalman, and particle filters were studied in [15]–[17], respectively.

Unfortunately, unlike its discrete counterpart, there are no existing works on the distributed continuous-discrete Bayesian filtering. That means modeling and solving the asynchronous cooperative problem in the continuous-discrete filtering framework is challenging.

C. MAIN CONTRIBUTIONS

In this work, for the first time we establish the mathematical model of the asynchronous cooperative localization problem for mobile nodes, and solve this problem by our proposed distributed prior-cut algorithm. *Two key features of our algorithms are: 1) The distributed prior-cut algorithm does not need any synchronization for the communication links and measurements processes in the entire wireless network. 2) Different from the existing localization algorithms, the distributed prior-cut algorithm does not require any scheduled iteration for exchanging local information, and the localization is conducted in a timely and fully distributed manner.*

Before giving the detailed contributions, we highlight the key idea of distributed prior-cut algorithm here. Different from the BP method which marginalizes the posterior (see Section I-B), the distributed prior-cut algorithm is based on cutting the prior into two parts at each node: one part is closely related to the self-location, and the other part is not. Then, every node refines its prior from the received messages by prior cutting, and discards less useful information accordingly (i.e. only collecting the useful information for self-localization).

The main contributions are given as follows:

- **Asynchronous model.** The asynchronous cooperative localization problem is modeled in the continuous-time domain. To be more specific, the mobility models of mobile nodes are modeled by stochastic differential equations, where a measurement can be taken and a communication can happen at arbitrary time instants without any time-slotted constraint.
- **Centralized algorithm analysis.** We give the centralized localization algorithm by the continuous-discrete Bayesian filtering theory, where the global information in the entire network is utilized. Then, we strictly prove an important property of the centralized algorithm that: if we cut the prior properly, i.e. the cut two parts are independent enough, then a localization algorithm with partial information can perform very close to the centralized algorithm.
- **Distributed algorithm design.** By the analysis of the centralized localization algorithm, the distributed prior-cut algorithm is proposed. This algorithm has a database at each node to memorize the history information which is different from the existing algorithms in the Bayesian framework. When a node receives the related information from other nodes (information collecting), the database update is triggered (information inferring). Specifically, the history information is updated by the continuous-discrete Bayesian filter, and the priors are refined by prior cutting which can significant reduce the computation complexity and communication overhead in estimating locations. After updating the database, the node broadcasts its refined information to other

TABLE 1. List of main symbols.

Symbol(s)	Description
$\mathcal{I}$, $\mathcal{A}$, $\mathcal{L}$	Sets of all nodes, anchor nodes, and non-anchor nodes
$x_i(t)$, $y_i(t)$	System state and location of node $i \in \mathcal{I}$ at time t
$z_{i,j}(t^{\mathrm{ms}}_{i,j,k})$	The k^{th} measurement that node i measures the distance between nodes i and j (i.e. measurement $i \leftarrow j$)
$Z_{i,j}[t]$	Set of all the measurements $i \leftarrow j$ within time window $[0,\ t]$
$Z_i[t]$	Set of all the measurements $i \leftarrow \cdot$ within time window $[0,\ t]$
$Z[t]$	Set of all measurements in the entire network within time window $[0,\ t]$
$m^{\mathrm{bc}}_{l,k}$	The k^{th} broadcast message by non-anchor node $l \in \mathcal{L}$ (anchor nodes do not broadcast messages)
$m^{\mathrm{rc}}_{l,j}(t^{\mathrm{rc}}_{l,j,k})$	The k^{th} received message at non-anchor node $l \in \mathcal{L}$ from non-anchor node $j \in \mathcal{L}$ (anchor nodes do not receive broadcast messages from non-anchor nodes)
$M^{\mathrm{rc}}_{l,j}[t]$	Set of all the messages received by non-anchor node $l \in \mathcal{L}$ from non-anchor node $j \in \mathcal{L}$ within time window $[0,\ t]$
$M^{\mathrm{rc}}_{l}[t]$	Set of all the messages received by non-anchor node $l \in \mathcal{L}$ during $[0,\ t]$
$x^*_a(t^{\mathrm{ms}}_{a,l,k})$	Anchor state (i.e. the system state) of anchor node a at measurement time $t^{\mathrm{ms}}_{a,l,k}$
$W_{l,a}[t]$	Set of all the anchor states received by non-anchor node $l \in \mathcal{L}$ from anchor node $a \in \mathcal{A}$ during $[0,\ t]$
$W_l[t]$	Set of all the anchor states received by non-anchor node $l \in \mathcal{L}$ during $[0,\ t]$
$W[t]$	Set of all the received anchor states in the entire network during $[0,\ t]$
$\hat{p}_l(\cdot)$	Local probability deduced from the perspective of non-anchor node $l \in \mathcal{L}$
$\hat{y}_l(t)$	Location estimation of node $l \in \mathcal{L}$ at time t
$\mathcal{G}^{\mathrm{ms}}_k$	The measurement graph of measurement set $Z[t^{\mathrm{ms}}_k]$
$\mathcal{C}^{\mathrm{ms}}_{k,r}$	The r^{th} measurement cluster in $\mathcal{G}^{\mathrm{ms}}_k$
$\mathcal{B}_l(t)$	The database at non-anchor node $l \in \mathcal{L}$ which is updated at every $t^{\mathrm{upd}}_{l,k}$ (we define $\mathcal{B}_{l,k} := \mathcal{B}_l(t^{\mathrm{upd}}_{l,k})$)

nodes (information sharing), but only the nodes who determine this information is useful, will use this information. Each node uses the updated information to estimate its own location.

Our simulation results show that the proposed distributed prior-cut algorithm performs nearly the same as the centralized algorithm.

D. PAPER ORGANIZATION

In Section II, the system model (including node mobility model and information exchanging model) is given, where the measurements and communications are asynchronous. Then the cooperative localization framework is provided and the cooperative localization problem is formulated in Section III. In Section IV, we propose and analyze the centralized localization algorithm. Base on the analysis in Section IV, we design the distributed prior-cut algorithm in Section V. In Section VI, simulation results are shown to illustrate the effectiveness of the distributed prior-cut algorithm, where two cooperative localization case studies for the mobile users and Unmanned Aerial Vehicle (UAV) are presented, respectively. Finally, the concluding remarks are given in Section VII.

E. NOTATION

Throughout this paper, $\mathbb{R}$, $\overline{\mathbb{R}}_+$ and $\mathbb{Z}_+$ denote the set of real numbers, non-negative real numbers, and positive integers, respectively, and $\mathbb{R}^n$ stands for n-dimensional Euclidean space. We use x represent a random variable or its realization. For a random variable/vector x, $p(x)$ denotes its pdf (probability density function), and supp$[x]$ returns its support, i.e. supp$[x] = \{x\colon p(x) \neq 0\}$. The mathematical statement $x \in \mathcal{X}$ means supp$[x] = \mathcal{X}$. $\{x(t)\}_{t\in\mathcal{T}}$ represents a stochastic process, and $\mathrm{d}x(t)/\mathrm{d}t$ denotes its time derivative at t in the mean square sense [18]. We use $\|\cdot\|_q$ to denote the q-norm or the $\mathcal{L}^q$-norm, and $\|\cdot\|$ to represent the Euclidean norm or the $\mathcal{L}^2$-norm. We also give a list of main symbols in this paper, which is shown in Table 1.

II. SYSTEM MODEL

Consider a wireless network with I nodes. For each node $i \in \{1,\ldots,I\} =: \mathcal{I}$, its location trajectory $y_i(t)$ follows a stochastic process $\{y_i(t)\}_{t\in\mathcal{T}}$ ($\mathcal{T} := \overline{\mathbb{R}}_+$) described by a state-space model:

$$\mathrm{d}x_i(t) = f_i(x_i(t), t)\mathrm{d}t + \mathrm{d}\beta_i(t), \tag{1}$$

$$y_i(t) = g_i(x_i(t)), \tag{2}$$

where (1) and (2) are called the state equation and the location equation, respectively. The state equation is a stochastic differential equation, and $t \in \mathcal{T}$ denotes the global time. $x_i(t) \in \mathcal{X}_i = \mathbb{R}^{n_i}$ stands for the location-related state (or system state) of node i, which can contain the location, velocity, acceleration, etc. We assume the initial condition $x_i(0)$ for different $i \in \mathcal{I}$ are mutually independent. The nonlinear map $f_i : \mathcal{X}_i \times \mathcal{T} \to \mathcal{X}_i$ is the system function, and it captures the key feature for how the system state evolves with time (a simple example is given in Remark 1). $\{\beta_i(t)\}_{t\in\mathcal{T}}$ is a n_i-dimensional Brownian motion with diffusion matrix $Q_i \in \mathbb{R}^{n_i\times n_i}$,[1] which

[1] That means $\mathrm{d}\beta_i(t)$ satisfies $\mathbb{E}[\mathrm{d}\beta_i(t)\mathrm{d}\beta_i^T(t)] = Q_i\mathrm{d}t$.

describes the additive noise for state evolutions. We assume $\{\beta_i(t)\}_{t\in\mathcal{T}}$ $(i \in \mathcal{I})$ are mutually independent. The location equation determines the node location $y_i(t) \in \mathcal{Y}_i \subseteq \mathbb{R}^d$ $(d \in \{1, 2, 3\})$, where $g_i : \mathcal{X}_i \to \mathcal{Y}_i$ is the location function, which reflects the relationship between the state and location. Note that if $x_i(t)$ refers to the location of the node i, then $g_i(x_i(t)) = x_i(t)$ is the identity map.

Remark 1: A simple example for the mobility model described by (1) *and* (2) *is a 1-dimensional Brownian motion, where $x_i(t)$ gives the location of node i, and*

$$f_i(x_i(t), t) = 0, \quad g_i(x_i(t)) = x_i(t), \quad Q_i = 1, \quad i \in \mathcal{I}.$$

The initial condition is $x_i(0) \sim \mathrm{N}(0, 1)$, *i.e. it follows the standard Gaussian distribution. Thus, at each time t, location $y_i(t)$ is a realization of $y_i(t) \sim \mathrm{N}(0, t + 1)$. Note that the Brownian motion is a commonly used model for describing human mobilities in wireless networks (see [19]). Our mobility model not only embraces the standard Brownian motion as a special case but also describes more complicated and realistic mobilities [20], e.g., for Unmanned Aerial Vehicles (UAVs) in Section VI-B.*

For localization problems, location $y_i(t)$ is usually not obtainable by node i itself, except for anchor nodes $i \in \mathcal{A}$, where $\mathcal{A}$ is called the anchor set. For the other nodes $i \in \mathcal{I} \setminus \mathcal{A} =: \mathcal{L}$, they are unable to get their locations $y_i(t)$, and we call them non-anchor nodes. Without loss of generality, we set $\mathcal{L} = \{1, \ldots, L\}$ and $\mathcal{A} = \{L + 1, \ldots, I\}$. We assume that anchor node $a \in \mathcal{A}$ can accurately get its true state and location at any given time $t \in \mathcal{T}$, which are labeled as $x_a^*(t)$ and $y_a^*(t) = g_a(x_a^*(t))$, respectively. Note that for each anchor node $a \in \mathcal{A}$, its state $x_a(t)$ of (1) is degenerated to a deterministic vector; but for each non-anchor node $l \in \mathcal{I} \setminus \mathcal{A} =: \mathcal{L}$, it does not know its exact state or location by itself, and the only way to estimate its own location is to cooperate with other nodes.

In this work, the cooperation is based on exchanging two kinds of information: the direct information (see Section II-A) and the indirect information (see Section II-B). Briefly speaking, the direct information refers to the measurements directly taken by a node, and it introduce new location-related information in the whole network. For the indirect information, it does not generate new information in the whole network, but helps all nodes to share their information to each other through communications.

A. DIRECT INFORMATION

The direct information exchanged between nodes is the distance-based measurements directly taken by a node. For example, node i can take a measurement of the distance between node j and node i by using the TOA or TDOA estimation. The detailed description is given as follows.

At time $t^{\mathrm{ms}}_{i,j,k} \in \{t^{\mathrm{ms}}_{i,j,k} \in \mathcal{T} \setminus \{0\} : k \in \mathcal{K}^{\mathrm{ms}}_{i,j} \subseteq \mathbb{Z}_+\} =: \mathcal{T}^{\mathrm{ms}}_{i,j}$ $(j \in \mathcal{I})$, node $i \in \mathcal{I}$ measures the distance between nodes j and i via

$$z_{i,j}(t^{\mathrm{ms}}_{i,j,k}) = \|y_i(t^{\mathrm{ms}}_{i,j,k}) - y_j(t^{\mathrm{ms}}_{i,j,k})\| + v_{i,j}(t^{\mathrm{ms}}_{i,j,k}). \tag{3}$$

Equation (3) is called the measurement equation, and $z_{i,j}(t^{\mathrm{ms}}_{i,j,k}) \in \mathcal{Z}_{i,j} \subseteq \mathbb{R}$ is called the measurement $i \leftarrow j$ at time $t^{\mathrm{ms}}_{i,j,k}$. Random variable $v_{i,j}(t^{\mathrm{ms}}_{i,j,k})$ represents the measurement noise and the process $\{v_{i,j}(t^{\mathrm{ms}}_{i,j,k})\}_{k\in\mathcal{K}^{\mathrm{ms}}_{i,j}}$ is white.[2] The measurement equation in (3) is a general model for a variety of distance-based measurement methods. The specific measurement method is not the focus of this work. In this paper, we assume that $x_i(0)$, $\{\beta_i(t)\}_{t\in\mathcal{T}}$, and $\{v_{i,j}(t^{\mathrm{ms}}_{i,j,k})\}_{k\in\mathcal{K}^{\mathrm{ms}}_{i,j}}$ are mutually independent. We can see that $t^{\mathrm{ms}}_{i,j,k}$ for different node i are not required to be synchronous.

The assumption of taking internodal measurements is given as follows.

Assumption 1: No measurements are taken between any two anchor nodes.

The measurements can be taken between two non-anchor nodes, or one non-anchor node and one anchor node. However, for anchor nodes, since the states and the locations are exactly known, there is no need to take measurements between anchor nodes. Thus, Assumption 1 is reasonable.

An example for measurements is given in Fig. 1.

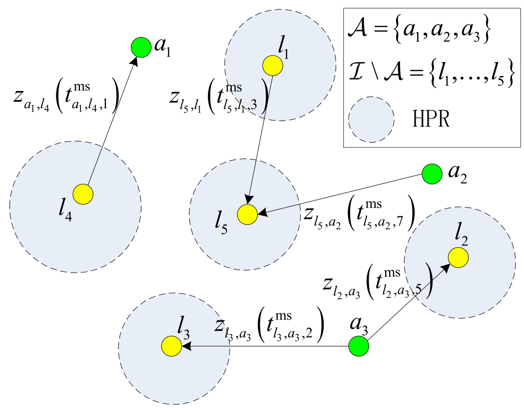


FIGURE 1. Illustrations of direct information. There are 8 nodes in this network, where points a_1, a_2, a_3 stand for the anchor nodes, and points $l_1, \ldots, l_5$ denote the non-anchor nodes. The shaded circles with dashed boundary are high probability regions (HPRs) for nodes $l_1, \ldots, l_5$, where an HPR for a node is the region that the true location falls in with high probability. For anchor nodes a_1, a_2, a_3, they know their own true locations. The straight arrows $l_1 \to l_5$, $a_2 \to l_5$, $l_4 \to a_1$, $a_3 \to l_3$, and $a_3 \to l_2$ represent the measurements $z_{l_5,l_1}(t^{\mathrm{ms}}_{l_5,l_1,3})$, $z_{l_5,a_2}(t^{\mathrm{ms}}_{l_5,a_2,7})$, $z_{a_1,l_4}(t^{\mathrm{ms}}_{a_1,l_4,1})$, $z_{l_3,a_3}(t^{\mathrm{ms}}_{l_3,a_3,2})$, and $z_{l_2,a_3}(t^{\mathrm{ms}}_{l_2,a_3,5})$ at time instants $t^{\mathrm{ms}}_{l_5,l_1,3}$, $t^{\mathrm{ms}}_{l_5,a_2,7}$, $t^{\mathrm{ms}}_{a_1,l_4,1}$, $t^{\mathrm{ms}}_{l_4,l_3,1}$, $t^{\mathrm{ms}}_{l_3,a_3,2}$, and $t^{\mathrm{ms}}_{l_2,a_3,5}$, respectively.

For the internodal measurements, we provide some notations for the convenience of discussion in the rest of the paper. At time t, the set of all the measurements $i \leftarrow j$ during $[0,\ t]$ is labeled as $Z_{i,j}[t] := \{z_{i,j}(t^{\mathrm{ms}}_{i,j,k}) : t^{\mathrm{ms}}_{i,j,k} \in \mathcal{T}^{\mathrm{ms}}_{i,j}[t]\}$, where $\mathcal{T}^{\mathrm{ms}}_{i,j}[t] := \mathcal{T}^{\mathrm{ms}}_{i,j} \bigcap [0,\ t]$, i.e. $\mathcal{T}^{\mathrm{ms}}_{i,j}[t]$ is the set of measurement time instants (in $\mathcal{T}^{\mathrm{ms}}_{i,j}$) within time window $[0,\ t]$. We define $Z_i[t] =: \bigcup_{j\in\mathcal{I}} Z_{i,j}[t]$ which stands for the set of all the measurements $i \leftarrow \cdot$, i.e. taken by node i, during $[0,\ t]$. Furthermore, we define $Z[t] := \bigcup_{i\in\mathcal{I}} Z_i[t]$ to represent the

[2] It means $v_{i,j}(t^{\mathrm{ms}}_{i,j,k})$ $(t^{\mathrm{ms}}_{i,j,k} \in \mathcal{T}^{\mathrm{ms}}_{i,j})$ are mutually independent and have zero mean.

set of all the measurements in the whole network within time window $[0, t]$.

B. INDIRECT INFORMATION

The indirect information refers to the messages exchanged between non-anchor nodes. A message can include the measurements and other information related to the nodal locations. For example, after non-anchor node l obtaining direct information $z_{l,j}(t^{\mathrm{ms}}_{i,j,k})$ from node $j \in \mathcal{I}$, it can transmit this measurement as a message (indirect information) to other non-anchor nodes. Non-anchor node l can include its location information $y_l(t^{\mathrm{ms}}_{l,j,k})$ as well to the message to help other non-anchor nodes' localization.

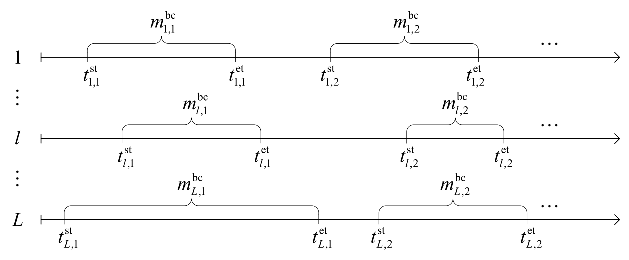


FIGURE 2. Illustrations of broadcasting time.

In this work, each non-anchor node $l \in \mathcal{L}$ transmits messages to other nodes by broadcasting, where the broadcasting time instants are illustrated in Fig. 2. In this figure, $m^{\mathrm{bc}}_{l,k}$ refers to message $k \in \mathcal{K}^{\mathrm{bc}}_l$ broadcasted by non-anchor node l, and the broadcast starts from $t^{\mathrm{st}}_{l,k}$ and ends at $t^{\mathrm{et}}_{l,k}$. When receiving the broadcast messages, each non-anchor node spends time to finish decoding and get the message. This is illustrated in Fig. 3, where broadcast messages $m^{\mathrm{bc}}_{l,1}$ and $m^{\mathrm{bc}}_{l,2}$ are received by non-anchor node $j \neq l$ ($j \in \mathcal{L}$) at time instants $t_{j,l,1}$ and $t_{j,l,2}$, respectively. For general cases, at time $t^{\mathrm{rc}}_{l,j,k} \in \{t^{\mathrm{rc}}_{l,j,k} \in \mathcal{T} : k \in \mathcal{K}^{\mathrm{rc}}_{l,j} \subseteq \mathbb{Z}_+\} =: \mathcal{T}^{\mathrm{rc}}_{l,j}$ ($l, j \in \mathcal{L}$), non-anchor node l receives message $m^{\mathrm{rc}}_{l,j}(t^{\mathrm{rc}}_{l,j,k})$ from non-anchor node j. Note that not all the broadcast message can be successfully received by node l, because the communication channel is not always in good condition. We can see that this communication framework does not need any synchronization.

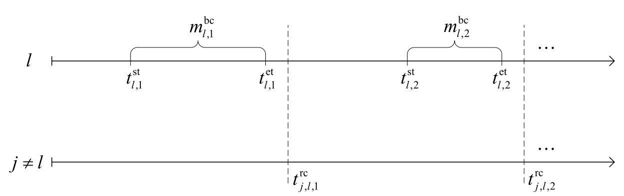


FIGURE 3. Illustrations of receiving time.

Similar to Section II-A, we give some notations for the indirect information. $M^{\mathrm{rc}}_{l,j}[t] := \{m_{l,j}(t^{\mathrm{rc}}_{l,j,k}) : t^{\mathrm{rc}}_{l,j,k} \in \mathcal{T}^{\mathrm{rc}}_{l,j}[t]\}$ denotes the set of all the messages received by l from non-anchor node j within time window $[0, t]$, where $\mathcal{T}^{\mathrm{rc}}_{l,j}[t] := \mathcal{T}^{\mathrm{rc}}_{l,j} \bigcap [0, t]$. $M^{\mathrm{rc}}_l[t] := \bigcup_{j \in \mathcal{L}} M^{\mathrm{rc}}_{l,j}[t]$ represents the set of all the messages received by non-anchor node l during $[0, t]$.

C. ASYNCHRONOUS LOCAL CLOCK

In Section II-A and Section II-B, even though the direct and indirect information is exchanging in an asynchronous manner, the time instants are still based on the global time system. But actually, each node deals with information based on its local clock. For example, time instant $t^{\mathrm{ms}}_{i,j,k}$ would be $t'^{\mathrm{ms}}_{i,j,k}$ at node 1, and $t''^{\mathrm{ms}}_{i,j,k}$ at node 2. However, for each node, it is easy to convert the local time from other nodes to its local time system, by simply adding the time difference. For example, if $t'^{\mathrm{ms}}_{i,j,k} = 1$ at node 1, while $t''^{\mathrm{ms}}_{i,j,k} = 2$ at node 2, then the time difference between the local clocks of 1 and 2 is $t'^{\mathrm{ms}}_{i,j,k} - t''^{\mathrm{ms}}_{i,j,k} = -1$, which means the clock at node 2 is 1 second faster than that at node 1, and all the time instants in the information from node 2 should be added with -1 at node 1. Note that this process does not need any synchronization in the entire network, and all the adjustments are done locally.

In this paper, we still use the global time system, since is helpful to formulate and discuss the problem. It should be also noted that our focus is the asynchronous measurements and communications, rather than simply the asynchronous local clocks.

III. COOPERATIVE LOCALIZATION FRAMEWORK AND DESIGN PROBLEM

A. COOPERATIVE LOCALIZATION FRAMEWORK

In this section, we describe the proposed cooperative localization framework and define the localization design problem. We consider a practical scenario that each node $l \in \mathcal{L}$ does not know when the next measurement $z_{l,j}(t^{\mathrm{ms}}_{l,j,k+1})$ ($j \in \mathcal{I}$) or the next message $m_{l,j}(t^{\mathrm{rc}}_{l,j,k+1})$ ($j \in \mathcal{L}$) will come,[3] and all the information one node can use is the received measurements and messages at hand.

To conduct the cooperative localization, all the nodes (including both anchor nodes and non-anchor nodes) work cooperatively.

1) ANCHOR NODES

- For each anchor node $a \in \mathcal{A}$, it can take measurement from non-anchor node $l \in \mathcal{L}$, say $z_{a,l}(t^{\mathrm{ms}}_{a,l,k})$. After taking this measurement, it transmits its state $x^*_a(t^{\mathrm{ms}}_{a,l,k})$ as well as the measurement $z_{a,l}(t^{\mathrm{ms}}_{a,l,k})$ to node l immediately.
- For each anchor node $a \in \mathcal{A}$, it can be measured by non-anchor node $l \in \mathcal{L}$,[4] and we label the corresponding measurement as $z_{l,a}(t^{\mathrm{ms}}_{l,a,k})$. When anchor node a realizes that it is being measured, it sends its state $x^*_a(t^{\mathrm{ms}}_{l,a,k})$ to node l at once.

Assumption 2: The time spent on transmission for $z_{a,l}(t^{\mathrm{ms}}_{a,l,k})$, $x^*_a(t^{\mathrm{ms}}_{a,l,k})$ and $x^*_a(t^{\mathrm{ms}}_{l,a,k})$ can be neglected.

[3] One reason for it is that the measurement/meassge cannot be always successfully taken/received.

[4] That means the distance between nodes a and l is measured by node l.

Since the anchor state $x_a^*(t_{a,l,k}^{\rm ms})$ or $x_a^*(t_{l,a,k}^{\rm ms})$ just contains one node's information, and $z_{a,l}(t_{a,l,k}^{\rm ms})$ is merely a one-dimensional number, the transmission can be completed timely. Hence, Assumption 2 is appropriate.

Similar to Section II-A and Section II-B, we provide some notations for the sets of received anchor states. For non-anchor node l, the set of all the received anchor states from anchor node a during $[0,\ t]$ is

$$W_{l,a}[t] := \{x_a(t_{l,a,k}^{\rm ms}) \colon t_{l,a,k}^{\rm ms} \in \mathcal{T}_{l,a}^{\rm ms}[t]\} \bigcup \{x_a(t_{a,l,k}^{\rm ms}) \colon t_{a,l,k}^{\rm ms} \in \mathcal{T}_{a,l}^{\rm ms}[t]\},$$

where $\mathcal{T}_{l,a}^{\rm ms}[t]$ and $\mathcal{T}_{a,l}^{\rm ms}[t]$ correspond to the sets of measurement times (see Section II-A), since the anchor states are sent when anchor node a takes measurements or is measured. We define $W_l[t] := \bigcup_{a\in\mathcal{A}} W_{l,a}[t]$ as the set of all the received anchor states at non-anchor node l within time window $[0,\ t]$. Furthermore, $W[t] := \bigcup_{l\in\mathcal{L}} W_l[t]$ stands for the set of all the received anchor states in the whole network during $[0,\ t]$.

2) NON-ANCHOR NODES

For each non-anchor node $l \in \mathcal{L}$, it wants to estimate its location $y_l(t)$ for the current time instant t with the help of the other nodes. This is called the cooperative localization, which is briefly illustrated in Fig. 4.

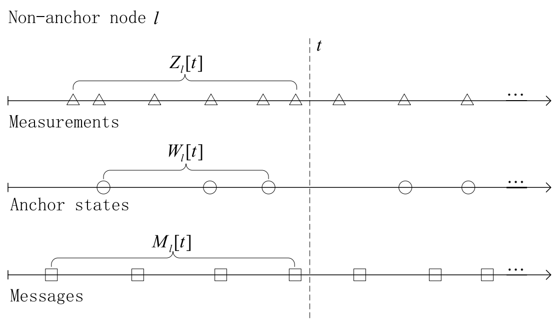


FIGURE 4. Illustrations of cooperative localization at non-anchor node $l \in \mathcal{L}$. The triangles, circles, and squares represent the measurements, anchor states, and messages, respectively. Note that at time t, node l utilizes the information from $Z_l[t]$, $W_l[t]$, and $M_l[t]$ to estimate the location $y_l(t)$.

From Fig. 4, we can see that all the information a non-anchor node $l \in \mathcal{L}$ can obtain by time instant t is from $Z_l[t]$, $W_l[t]$, and $M_l[t]$, and we call it the external information. As a results, different cooperative localization algorithms are indeed based on utilizing different parts of the external information.

B. COOPERATIVE LOCALIZATION DESIGN PROBLEM

To give a formal description of the cooperative localization problem, we need to introduce a concept – local probability, which is given as follows. Let $X(t) := \left[x_1^{\rm T}(t), \ldots, x_L^{\rm T}(t)\right]^{\rm T}$, i.e. the vector of all non-anchor nodes' states at time t. For the simplicity of analysis, we rewrite $X(t)$ as $\left[x_l^{\rm T}(t)\right]_{l\in\mathcal{L}}^{\rm T}$. Hence, a sub-vector of $X(t)$ with part of non-anchor nodes can be simply expressed as $\left[x_l^{\rm T}(t)\right]_{l\in\bar{\mathcal{L}}}^{\rm T} =: \bar{X}(t)$, where $\bar{\mathcal{L}} \subseteq \mathcal{L}$. Due to the limited communication bandwidth, global information $X(t)$ can hardly be exchanged among all non-anchor nodes. As a result, each non-anchor node $l \in \mathcal{L}$ at time $t \in \mathcal{T}$ can only know a portion of the global information as denoted by $\bar{X}_l(t)$ based on its received measurements and messages during $[0,\ t]$, where the subscript l is employed to distinguish the different sub-vectors $\bar{X}(t)$ for different non-anchor nodes. In this paper, the inference of $\bar{X}_l(t)$ refers to deduce the conditional probability $p(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$ which is conditioned on the received measurements, anchor states, and messages. Since the storage and computational capability of node l are limited, it is impossible to utilize all the information from $Z_l[t]$, $W_l[t]$, and $M_l[t]$ when t goes to sufficiently large.[5] That means only a portion of $Z_l[t]$, $W_l[t]$, and $M_l[t]$ can be utilized to infer $\bar{X}_l(t)$ at node l. To represent this "local" inference of $\bar{X}_l(t)$, we use the notation $\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$ [called the local probability of $\bar{X}_l(t)$], where $\hat{p}_l(\cdot)$ means it is deduced from the perspective of node l, and the local probability must be properly designed at node l to largely approximate $p(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$. As a special case of local probabilities, the initial condition of non-anchor node l is $\hat{p}_l(x_l(0))$ which reflects the initial guess of state $x_l(0)$ and is only known to node i itself, while the initial condition $\hat{p}_l(x_j(0))$ is not available (undefined) for other node $j \neq l$ (where $j \in \mathcal{L}$). Note that local probabilities are the first part to be designed in our cooperative localization algorithm.

With local probabilities, we can define the local location estimation $\hat{y}_l(t)$ from the perspective of non-anchor nodes $l \in \mathcal{L}$ as follows:

$$\begin{aligned}\hat{y}_l(t) &= \hat{\mathbb{E}}_{y_l(t)}^{\bar{X}_l(t)}[y_l(t)|Z_l[t], W_l[t], M_l[t]] \\ &= \int_{\bar{\mathcal{X}}_l(t)} g_l(x_l(t))\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])\mathrm{d}\bar{X}_l(t),\end{aligned} \tag{4}$$

where the superscript $\bar{X}_l(t)$ of $\hat{\mathbb{E}}[\cdot]$ means the estimation is based on the local probability of $\bar{X}_l(t)$, and $\bar{\mathcal{X}}_l(t) \ni \bar{X}_l(t)$ is the support of $\bar{X}_l(t)$. In this paper, we assume

$$\int_{\bar{\mathcal{X}}_l(t)} \|g_l(x_l(t))\|\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])\mathrm{d}\bar{X}_l(t) < \infty.$$

The second part to be designed is what to exchange between different nodes, i.e. designing the broadcast messages. The broadcast messages can include the measurements and local probabilities from other nodes, or any useful information for location estimation. Note that whether these messages are successfully transmitted is dependent on the physical layer and the MAC layer of the network which are considered in Section VI.

Now, it is readily to formally introduce the cooperative localization problem.

Problem 1 (Cooperative Localization): At time t, each non-anchor node $l \in \mathcal{L}$ calculates the location estimation

[5]One important reason is that the Markov property cannot be guaranteed when the information is collected in a distributed manner.

$\hat{y}_l(t)$ by (4)*, where the prior $\hat{p}(x_l(0))$, the measurement set $Z_l[t]$, the anchor state set $W_l[t]$, and the message set $M_l[t]$ are known to node l. With the help of anchor nodes (see Section III-A1), a cooperative localization algorithm for each non-anchor node l is to:*

i) *design the local probability $\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$ at node $l \in \mathcal{L}$;*
ii) *design the broadcast message $m^{\text{bc}}_{l,k}$ ($k \in \mathcal{K}^{\text{bc}}_l$) for each node $l \in \mathcal{L}$;*

such that the mean squared error $\mathbb{E}\|\hat{y}_l(t) - y_l(t)\|^2$ is minimized at each $t \in \mathcal{T}$.

If we do not consider the communication constraints, the local probability at each non-anchor node $l \in \mathcal{L}$ will become

$$\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t]) = p(X(t)|Z[t], W[t]),$$

i.e. it contains the global information since ideal communications can make $Z[t]$ and $W[t]$ known to every non-anchor node. In this case, the solution to the cooperative localization in Problem 1 is equivalent to run the centralized localization algorithm at each non-anchor node where the global information can be always obtained in estimating locations. This centralized algorithm is given in Lemma 1 in Section IV-A. Note that the centralized algorithm can serves as a benchmark for distributed algorithms in which the communication constraints are considered.

Since the optimal solution to Problem 1 is hard to derive when communications constraints are considered, in this work, we are aim to find a suboptimal solution to Problem 1 by analyzing the performance gap between the distributed and centralized algorithms. The analysis of performance gap is given in Section IV, and the distributed algorithm is proposed in Section V.

IV. INSPIRATION FROM CENTRALIZED LOCALIZATION

In this section, we propose a centralized localization algorithm, as a benchmark for distributed algorithms, which can utilize all the states of nodes and all the measurements in the whole network (see Section IV-A), and then show that the centralized localization algorithm has a key property which is very helpful in the designs of cooperative algorithms (see Section IV-B).

A. CENTRALIZED LOCALIZATION ALGORITHM

The centralized localization algorithm is to derive the location estimation in a centralized manner. That means the set of all measurements $Z[t] = Z[t^{\text{ms}}_k]$ and the set of all anchor states $W[t] = W[t^{\text{ms}}_k]$ in this network are used, where t^{ms}_k is the largest time instant in $\mathcal{T}^{\text{ms}} := \bigcup_{i,j \in \mathcal{I}} \mathcal{T}^{\text{ms}}_{i,j}$ up to time t.[6] The centralized location estimation $\hat{y}^*_l(t)$ is given by

$$\begin{aligned}\hat{y}^*_l(t) &= \mathbb{E}_{y_l(t)}[y_l(t)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k]] \\ &= \int_{\mathcal{X}} g_l(x_l(t))p(X(t)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k])\text{d}X(t), \quad (5)\end{aligned}$$

where $l \in \mathcal{L}$ and $X(t) \in \mathcal{X}$. In this work, we assume

$$\int_{\mathcal{X}} \|g_l(x_l(t))\|p(X(t)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k])\text{d}X(t) < \infty. \quad (6)$$

The centralized algorithm can be implemented in the recursive Bayesian filtering framework as shown in Lemma 1.

Lemma 1 (Centralized Localization Algorithm): At time instant t, the estimated location $\hat{y}^*_l(t)$ for non-anchor node $l \in \mathcal{L}$ is calculated by (5)*, where $p(X(t)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k])$ is derived by*

$$\int_{\mathcal{X}} p(X(t)|X(t^{\text{ms}}_k))p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k])\text{d}X(t^{\text{ms}}_k), \quad (7)$$

in which $p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_k])$ is computed by the following recursive equations:

- ***Initialization.** The recursion starts from the prior distribution $p(X(0)) = \prod_{l \in \mathcal{L}} \hat{p}_l(x_l(0))$.*
- ***Prediction.** At time instant $t^{\text{ms}}_k \in \mathcal{T}^{\text{ms}}$, the prior distribution $p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_{k-1}], W[t^{\text{ms}}_{k-1}])$ is derived by the Chapman-Kolmogorov equation*

$$\begin{aligned}\int_{\mathcal{X}} & p(X(t^{\text{ms}}_k)|X(t^{\text{ms}}_{k-1})) \\ & \cdot p(X(t^{\text{ms}}_{k-1})|Z[t^{\text{ms}}_{k-1}], W[t^{\text{ms}}_{k-1}])\text{d}X(t^{\text{ms}}_{k-1}), \quad (8)\end{aligned}$$

where $t^{\text{ms}}_0 := 0$ and $p(X(0)|Z[0], W[0]) := p(X(0))$.

- ***Update.** Given the measurement set and the received anchor state set at t^{ms}_k, i.e. $Z(t^{\text{ms}}_k) := Z[t^{\text{ms}}_k] \setminus Z[t^{\text{ms}}_{k-1}]$ and $W(t^{\text{ms}}_k) := W[t^{\text{ms}}_k] \setminus W[t^{\text{ms}}_{k-1}]$, respectively, the posterior distribution $p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_k], W[t^{\text{ms}}_k])$ can be computed by*

$$\frac{p(Z(t^{\text{ms}}_k)|X(t^{\text{ms}}_k), W(t^{\text{ms}}_k))p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_{k-1}], W[t^{\text{ms}}_{k-1}])}{\text{Normalization}}, \quad (9)$$

where

$$\begin{aligned}\text{Normalization} = \int_{\mathcal{X}} & p(Z(t^{\text{ms}}_k)|X(t^{\text{ms}}_k), W(t^{\text{ms}}_k)) \\ & \cdot p(X(t^{\text{ms}}_k)|Z[t^{\text{ms}}_{k-1}], W[t^{\text{ms}}_{k-1}])\text{d}X(t^{\text{ms}}_k),\end{aligned}$$

and the likelihood function $p(Z(t^{\text{ms}}_k)|X(t^{\text{ms}}_k), W(t^{\text{ms}}_k))$ is derived from the measurement equation (3).

Proof: See Appendix A. ■

Remark 2: In Lemma 1, all the probabilities are defined except for $p(X(t)|X(t^{\text{ms}}_k))$ in (7) *and $p(X(t^{\text{ms}}_k)|X(t^{\text{ms}}_{k-1}))$ in* (8)*. These two probabilities can be derived by solving the well-known Fokker-Planck equation (see Chapter 4.9 in [14]) of the group of state equations* (1) *for $l \in \mathcal{L}$, when the system function is linear w.r.t. the system state $x_l(t)$. If the system function is nonlinear, solving the Fokker-Planck equation is difficult, but we can numerically solve the state equations with different initial values instead.*[7] *These solutions can work as the propagated sigma points in Gaussian filters [6], [16], or the particles in particle filters [22], [23].*

[6]The subscript k in t^{ms}_k means that t^{ms}_k is the k^{th} smallest element in $\mathcal{T}^{\text{ms}}$.

[7]For example, we can use the Euler-Maruyama method or high-order methods [16], [21]–[23] to solve the state equations with initial values at t^{ms}_k for deriving $p(X(t)|X(t^{\text{ms}}_k))$.

From Lemma 1, we can see that when new measurements come, the algorithm should make the prediction [through (8)] and update [through (9)] to derive the posterior distribution $p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}])$. Thus, the distribution $p(X(t)|Z[t], W[t])$ is derived by (7), and the estimated location is $\hat{y}_l^*(t)$ is calculated by (5).

It is, $Z(t_k^{\text{ms}})$, the set of all the new measurements who updates the estimated location $\hat{y}_l^*(t) = \mathbb{E}_{y_l(t)}[y_l(t)|Z[t], W[t]]$. However, not all the new measurements and system states have notable contributions to the estimated location. Actually, we can utilize part of the measurements and the states without losing too much performance of location estimation. This property will be analyzed comprehensively in Section IV-B, and we stress that it plays a pivotal role in designing a cooperative localization algorithm, since not all the non-anchor nodes need to know all the new measurements and the states as the centralized algorithm does.

B. ESTIMATION GAP UNDER PARTIAL INFORMATION

In this subsection, we analyze the performance gap between the centralized location estimation and the location estimation when the information of measurements and anchor states is partially known (see Theorem 1), which plays a pivotal role in designing the cooperative localization algorithms.

Recall that the centralized location estimation

$$\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]]$$

is defined in (5). For the location estimation with partial information at t_k^{ms}, it has the following form

$$\begin{aligned}\mathbb{E}_{y_l(t)}^{\bar{X}(t)}&[y_l(t)|Z[t_{k-1}^{\text{ms}}], \bar{Z}(t_k^{\text{ms}}), W[t_{k-1}^{\text{ms}}], \bar{W}(t_k^{\text{ms}})]\\ &= \int_{\bar{\mathcal{X}}(t)} g_l(x_l(t)) p(\bar{X}(t)|Z[t_{k-1}^{\text{ms}}], \bar{Z}(t_k^{\text{ms}}), W[t_{k-1}^{\text{ms}}],\\ &\qquad \bar{W}(t_k^{\text{ms}}))\mathrm{d}\bar{X}(t), \qquad (10)\end{aligned}$$

where $\bar{Z}(t_k^{\text{ms}})$ and $\bar{W}(t_k^{\text{ms}})$ are the partial known measurement set and anchor set, respectively. Thus, the estimation gap is define as follows

$$\begin{aligned}\Big\| &\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]]\\ &- \mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\text{ms}}], \bar{Z}(t_k^{\text{ms}}), W[t_{k-1}^{\text{ms}}], \bar{W}(t_k^{\text{ms}})]\Big\|. \qquad (11)\end{aligned}$$

We can see that the smaller the estimation gap is, the closer the location estimation with partial information to the centralized location estimation will be.

Before analyzing the estimation gap, we propose two important concepts: the measurement graph and the measurement cluster, which are given in Definition 1.

Definition 1 (Measurement Graph and Measurement Cluster): At time instant t_k^{ms}, directed graph $\mathcal{G}_k^{\text{ms}} := \big(\mathcal{V}(Z(t_k^{\text{ms}})), \mathcal{E}(Z(t_k^{\text{ms}}))\big)$ is the measurement graph, where vertex set $\mathcal{V}(Z(t_k^{\text{ms}}))$ represents the set of nodes directly related to the measurements in $Z(t_k^{\text{ms}})$, and has the following form

$$\mathcal{V}(Z(t_k^{\text{ms}})) := \{i\colon z_{i,j}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}}) \vee z_{j,i}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}})\},$$

in which $\vee$ is the logical "or"; and edge set $\mathcal{E}(Z(t_k^{\text{ms}}))$ records the pairs of node related to the measurements which is

$$\mathcal{E}(Z(t_k^{\text{ms}})) := \{(j, i)\colon z_{i,j}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}}) \vee z_{j,i}(t_k^{\text{ms}}) \in Z(t_k^{\text{ms}})\}.$$

The measurement clusters are the connected components[8] of $\mathcal{G}_k^{\text{ms}}$ when ignoring the direction of all the edges. We label the measurement clusters as $\mathcal{C}_{r,k}^{\text{ms}}$, where $r \in \{1, \ldots, R_k\} =: \mathcal{R}_k$.

Remark 3: The measurement graph tells which nodes are related to the considered measurements (i.e. vertices), and how the measurements connect to these nodes (i.e. edges). An example of the measurement graph is given in Fig. 5, where we use the same network as that in Fig. 1. The vertices of $\mathcal{G}_{19}^{\text{ms}}$ are $\{a_1, a_2, l_1, l_4, l_5\}$ and the edges are $\{(l_4, a_1), (l_1, l_5), (a_2, l_5)\}$. Note that edge (i, j) corresponds to the measurement $z_{j,i}(t_{19}^{\text{ms}})$.

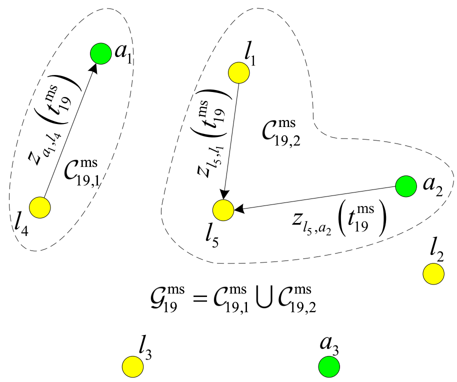


FIGURE 5. Illustration of measurement graph and measurement cluster. We continue using the network in Fig. 1, where $t_{19}^{\text{ms}} = t_{l_5,l_1,3}^{\text{ms}} = t_{l_5,a_2,7}^{\text{ms}} = t_{a_1,l_4,1}^{\text{ms}}$. The measurement graph is $\mathcal{G}_{19}^{\text{ms}} = (\{a_1, a_2, l_1, l_4, l_5\}, \{(l_4, a_1), (l_1, l_5), (a_2, l_5)\})$. Note that nodes a_3, l_2, and l_3 are not included, since their corresponding measurement times are not equal to t_{19}^{ms}. In $\mathcal{G}_{19}^{\text{ms}}$, there are two measurement clusters, i.e. $\mathcal{C}_{19,1}^{\text{ms}} = (\{a_1, l_4\}, \{(l_4, a_1)\})$ and $\mathcal{C}_{19,2}^{\text{ms}} = (\{a_2, l_1, l_5\}, \{(a_2, l_5), (l_1, l_5)\})$.

In terms of the measurement clusters, they reflect which nodes are directly connected by the measurements. Between any two clusters, there are no nodes linked by the measurements. From Fig. 5, we can see that the measurement graph $\mathcal{G}_{19}^{\text{ms}}$ contains two disjoint subgraphs $(\{a_1, l_4\}, \{(l_4, a_1)\})$ and $(\{a_2, l_1, l_5\}, \{(a_2, l_5), (l_1, l_5)\})$ which are exactly the measurement clusters $\mathcal{C}_{19,1}^{\text{ms}}$ and $\mathcal{C}_{19,2}^{\text{ms}}$, respectively.

With the measurement graph and measurement cluster, we define the admissible measurement-state (AMS) triple as follows.

Definition 2 (AMS Triple): Let $\bar{Z}(t_k^{\text{ms}}) \subseteq Z(t_k^{\text{ms}})$, $\bar{W}(t_k^{\text{ms}}) \subseteq W(t_k^{\text{ms}})$, and $\bar{X}(t_k^{\text{ms}}) \subseteq X(t_k^{\text{ms}})$. The measurement-state triple $\langle \bar{Z}(t_k^{\text{ms}}), \bar{X}(t_k^{\text{ms}}), \bar{W}(t_k^{\text{ms}})\rangle$

[8] A formal definition of connected component for an undirected graph is from [24] that: A path is a sequence of vertices where there is an edge connecting each vertex to the next vertex in the path. If there exists a path from vertex u to w, then we say that vertex w is reachable from vertex u. A connected component is a group of vertices in an undirected graph that are reachable from one another.

is admissible, if the following three conditions hold:

$$\big(\mathcal{V}(\bar{Z}(t_k^{\rm ms})), \mathcal{E}(\bar{Z}(t_k^{\rm ms}))\big) = \bigcup_{r\in\mathcal{R}'_k} \mathcal{C}_{k,r}^{\rm ms}, \tag{12}$$

$$\mathcal{V}(\bar{Z}(t_k^{\rm ms}))\bigcap \mathcal{L} \subseteq \mathcal{L}_{\bar{X}(t_k^{\rm ms})} \subseteq \mathcal{L}\setminus \mathcal{V}(\breve{Z}(t_k^{\rm ms})), \tag{13}$$

$$\mathcal{V}(\bar{Z}(t_k^{\rm ms}))\bigcap \mathcal{A} \subseteq \mathcal{A}_{W(t_k^{\rm ms})} \subseteq \mathcal{A}\setminus \mathcal{V}(\breve{Z}(t_k^{\rm ms})), \tag{14}$$

where $\mathcal{R}'_k \subseteq \mathcal{R}_k$, $\breve{Z}(t_k^{\rm ms}) = Z(t_k^{\rm ms}) \setminus \bar{Z}(t_k^{\rm ms})$, $\mathcal{L}_{\bar{X}(t_k^{\rm ms})} := \{l\colon x_l(t_k^{\rm ms}) \in \bar{X}(t_k^{\rm ms})\}$,[9] *and* $\mathcal{A}_{\bar{W}(t_k^{\rm ms})} := \{a\colon x_a(t_k^{\rm ms}) \in \bar{W}(t_k^{\rm ms})\}$.

Remark 4: The AMS triple contains three elements. The first one is the measurement set $\bar{Z}(t_k^{\rm ms})$ *which satisfies condition* (12). *This condition tells that graph* $\big(\mathcal{V}(\bar{Z}(t_k^{\rm ms})), \mathcal{E}(\bar{Z}(t_k^{\rm ms}))\big)$ *is exactly the union of connected components of graph* $\mathcal{G}_k^{\rm ms}$. *That means any node linked by* $\bar{Z}(t_k^{\rm ms})$ *cannot be connected by any other measurements in* $Z(t_k^{\rm ms})$. *For example, in Fig. 5,* $\bar{Z}(t_{19}^{\rm ms})$ *can be* $\{z_{a_1,l_4}(t_{19}^{\rm ms})\}$, $\{z_{l_5,l_1}(t_{19}^{\rm ms}), z_{l_5,a_2}(t_{19}^{\rm ms})\}$, *or* $\{z_{a_1,l_4}(t_{19}^{\rm ms}), z_{l_5,l_1}(t_{19}^{\rm ms}), z_{l_5,a_2}(t_{19}^{\rm ms})\}$.

The second element is the state set $\bar{X}(t_k^{\rm ms})$ *which satisfies condition* (13). *This condition states that the set of non-anchor nodes* $\mathcal{L}_{\bar{X}(t_k^{\rm ms})}$ *must contain non-anchor node set* $\mathcal{V}(\bar{Z}(t_k^{\rm ms}))\bigcap\mathcal{L}$ *and must be contained in non-anchor node set* $\mathcal{L}\setminus\mathcal{V}(\breve{Z}(t_k^{\rm ms}))$. *That means non-anchor node set* $\mathcal{L}_{\bar{X}(t_k^{\rm ms})}$ *must contain the non-anchor nodes linked by* $\bar{Z}(t_k^{\rm ms})$, *and cannot be associated with other measurements [i.e. measurements in* $\breve{Z}(t_k^{\rm ms})$*]. For example, in Fig. 5, if* $\bar{Z}(t_{19}^{\rm ms}) = \{z_{a_1,l_4}(t_{19}^{\rm ms})\}$, *then a valid* $\bar{X}(t_k^{\rm ms})$ *can be* $\{x_{l_3}(t_k^{\rm ms}), x_{l_4}(t_k^{\rm ms})\}$, *since condition* (13) *holds, i.e.*

$$\begin{aligned}\mathcal{V}(\bar{Z}(t_{19}^{\rm ms}))\bigcap\mathcal{L} &= \{l_4\} \subseteq \mathcal{L}_{\bar{X}(t_k^{\rm ms})}\\ &= \{l_3, l_4\} \subseteq \mathcal{L}\setminus\mathcal{V}(\breve{Z}(t_{19}^{\rm ms})) = \{l_2, l_3, l_4\}.\end{aligned}$$

However, for $t \in (t_{19}^{\rm ms},\ t_{20}^{\rm ms}]$, *state set* $\{x_{l_2}(t_k^{\rm ms})\}$ *is not a valid* $\bar{X}(t_k^{\rm ms})$, *since* $\mathcal{V}(\bar{Z}(t_{19}^{\rm ms}))\bigcap\mathcal{L} \nsubseteq \mathcal{L}_{\bar{X}(t_k^{\rm ms})}$ *[i.e.* $\{l_2\}$ *does not contain any non-anchor node linked by* $\bar{Z}(t_{19}^{\rm ms})$*]. Also, state set* $\{x_{l_1}(t_k^{\rm ms}), x_{l_3}(t_k^{\rm ms})\}$ *is not a valid* $\bar{X}(t_k^{\rm ms})$, *since* $\mathcal{L}_{\bar{X}(t_k^{\rm ms})} \nsubseteq \mathcal{L}\setminus\mathcal{V}(\breve{Z}(t_{19}^{\rm ms}))$ *[i.e.* $\{x_{l_1}(t_k^{\rm ms}), x_{l_3}(t_k^{\rm ms})\}$ *is associated with measurement* $z_{l_5,l_1}(t_{19}^{\rm ms}) \notin \bar{Z}(t_{19}^{\rm ms})$*)].*

The third element is the anchor state set $\bar{W}(t_k^{\rm ms})$ *satisfying condition* (14). *It says that anchor set* $\mathcal{A}_{W(t_k^{\rm ms})}$ *should include anchor set* $\mathcal{V}(\bar{Z}(t_k^{\rm ms}))\bigcap\mathcal{A}$. *This implies any anchor node associated with measurements in* $\bar{Z}(t_k^{\rm ms})$ *must be in set* $\mathcal{A}_{W(t_k^{\rm ms})}$. *For example, in Fig. 5, if we choose* $\bar{Z}(t_{19}^{\rm ms}) = \{z_{a_1,l_4}(t_{19}^{\rm ms})\}$, *then a valid* $W(t_k^{\rm ms})$ *can be* $\{a_1\}$ *or* $\{a_1, a_3\}$, *since it contains* $\mathcal{V}(\bar{Z}(t_{19}^{\rm ms}))\bigcap\mathcal{A} = \{a_1\}$ *and is included in* $\mathcal{A}\setminus\mathcal{V}(\breve{Z}(t_{19}^{\rm ms})) = \{a_1, a_3\}$.

Given prior $p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])$, each AMS triple determines a prior cut

$$p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])p(\breve{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]), \tag{15}$$

[9]Strictly speaking, $\mathcal{L}_{\bar{X}(t_k^{\rm ms})}$ refers to the set of nodes such that $[x_l(t_k^{\rm ms})]_{l\in\mathcal{L}_{\bar{X}(t_k^{\rm ms})}}$.

where $\breve{X}(t_k^{\rm ms}) = \Big[x_j^{\rm T}(t_k^{\rm ms})\Big]^{\rm T}_{j\in\mathcal{L}_{X(t_k^{\rm ms})}\setminus\mathcal{L}_{\bar{X}(t_k^{\rm ms})}}$. In (15), the first term and the second term are called the related sub-prior and unrelated sub-prior, respectively. The cut gap is defined as

$$\begin{aligned}\Big\| p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) &\\ - p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) &\\ \cdot p(\breve{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])\Big\|_\infty&,\end{aligned} \tag{16}$$

which measures the independence between the prior and prior cut. If the cut gap is small, then $\bar{X}(t_k^{\rm ms})$ and $\breve{X}(t_k^{\rm ms})$ tend to be independent of each other; and if the cut gap is zero, then they are independent. Theorem 1 says that the estimation gap defined in (11) is continuous with the cut gap, if the following two conditions are satisfied

$$\begin{aligned}\int_{\mathcal{X}} &\big\|\mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\rm ms})]\big\|\\ &\times p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))\mathrm{d}X(t_k^{\rm ms}) < \infty,\end{aligned} \tag{17}$$

$$\int_{\mathcal{X}} p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))\mathrm{d}X(t_k^{\rm ms}) < \infty, \tag{18}$$

where

$$\mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\rm ms})] = \int_{\mathcal{X}_l} g(x_l(t))p(x_l(t)|x_l(t_k^{\rm ms}))\mathrm{d}x_l(t). \tag{19}$$

Theorem 1 (Continuity of Estimation Gap): Let

$$\langle\bar{Z}(t_k^{\rm ms}), \bar{X}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms})\rangle$$

be an AMS triple, and conditions (17) *and* (18) *are satisfied. For node* $l \in \mathcal{L}_{\bar{X}(t_k^{\rm ms})} = \mathcal{L}_{\bar{X}(t)}$, $\forall\varepsilon > 0$, *there exists a* $\delta > 0$ *such that if the cut gap defined in* (16) *is not greater than* δ, *the the estimation gap defined in* (11) *is not greater than* ε.

Proof: See Appendix B. ■

Remark 5: The detailed relationship between δ *and* ε *is shown in equation* (51) *in Appendix B, which indicates the continuity of the estimation gap given in Theorem 1.*

Remark 6 (Implications from the Key Property in Theorem 1): Theorem 1 provides an important implication for designing cooperative localization algorithms that: If the cut gap is small (i.e. the two sub-priors in (15) *are nearly independent of each other), then the estimation gap in each node* $l \in \mathcal{L}_{\bar{X}(t)}$ *is also small. Also, it means the related nodes should share their information to estimate their locations cooperatively.*

V. DISTRIBUTED PRIOR-CUT ALGORITHM

In this section, we propose a cooperative localization algorithm termed the distributed prior-cut algorithm, where the prior refers to the local prior defined in Definition 3.

Before defining the local prior and posterior, we illustrate that in the distributed prior-cut algorithm, the local probability $\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$ is actually with the form $\hat{p}_l(\bar{X}_l(t)|\bar{Z}[t], \bar{W}[t])$, where $\bar{Z}[t] \subseteq Z[t]$ and $\bar{W}[t] \subseteq W[t]$. This is because, in the distributed prior-cut algorithm, the received messages during $[0,\ t]$ contain the information of the measurements and anchor states from other non-anchor nodes (see Section V-D).

Definition 3 (Local Prior and Posterior): For the local probability $\hat{p}_l(\bar{X}_l(t)|\bar{Z}[t], \bar{W}[t])$:

- *If all the measurements in $\bar{Z}[t]$ are within the time window $[0,\ t)$, i.e. not including t, then we label this measurement set as $\bar{Z}^{\mathrm{a}}[t]$. Similarly, $\bar{W}^{\mathrm{a}}[t]$ represents the anchor state set whose elements are within $[0,\ t)$. We call $\hat{p}_l(\bar{X}_l(t)|\bar{Z}^{\mathrm{a}}[t], \bar{W}^{\mathrm{a}}[t])$ as the local prior.*
- *If at least one measurement in $\bar{Z}[t]$ is at time instant t, then we label this measurement set as $\bar{Z}^{\mathrm{b}}[t]$. Likewise, $\bar{W}^{\mathrm{b}}[t]$ stands for the anchor state set which contains at least one anchor state at time instant t. We call $\hat{p}_l(\bar{X}_l(t)|\bar{Z}^{\mathrm{b}}[t], \bar{W}^{\mathrm{b}}[t])$ as the local posterior.*

A. METHODOLOGY

This subsection gives the main ideas of designing the distributed prior-cut algorithm mainly based on the main property (which is given in Theorem 1 and explained in Remark 6) derived in Section IV-B.

Firstly, the broadcast-message design in Section II is beneficial to the information sharing, since the information in one node is potentially useful for all nodes. Specifically, the more one node knows, the closer its local localization algorithm to the centralized algorithm (see Lemma 1) will be.

However, if all the nodes broadcast all their information without proper reduction, then the communication cost would be very large and cannot be afforded by a wireless communication network. Theorem 1 (see also Remark 6) provides an effective resolution to this problem: every node only needs to consider and transmit the messages related to the AMS triple (see Definition 2) with a small cut gap [defined in (16)]. It should be noted that the AMS triple used in each node is similar to the analysis in Section IV-B but based on the local information in each node, since the global information is usually unable to be accessed.

B. ALGORITHM STRUCTURE

In this subsection, we propose the structure the distributed prior-cut algorithm. The main ideas are:

- Each non-anchor node $l \in \mathcal{L}$ contains a database $\mathcal{B}_l(t)$ which contains all the knowledge on the whole network from the perspective of node l.
- After receiving measurements, anchor states, or messages, each non-anchor node l updates its database $\mathcal{B}_l(t)$ through these newly arrived information.
- Based on the newly updated database $\mathcal{B}_l(t)$, the broadcast messages are properly designed.
- Based on the newly updated database, each non-anchor node l estimates its location continuously.[10]

We can see that the database $\mathcal{B}_l(t)$ plays a pivotal role in our algorithm.

Now, we begin to propose the structure for the cooperative localization algorithm, which is given in Algorithm 1.

[10]Practically, the update of estimated location depends on the clock of the onboard chip.

Algorithm 1 Cooperative Localization Algorithm of Non-anchor Node l

1: **Initialization:** $\hat{p}_l(x_l(0))$.
2: **loop**
3:   **if** $Z_l(t) \neq \emptyset \ || \ W_l(t) \neq \emptyset \ || \ M_l(t) \neq \emptyset$ **then**
4:     Update the database $\mathcal{B}_l(t)$;
5:     Design broadcast messages;
6:   **end if**
7:   Calculate $\hat{p}_l(\bar{X}_l(t)|Z_l[t], W_l[t], M_l[t])$;
8:   Estimate $\hat{y}_l(t)$ by using (4);
9: **end loop**

In Algorithm 1, the initialization gives the local probability $\hat{p}_l(x_l(0))$ for non-anchor node l. Recall that at time $t = 0$, non-anchor node l does not know $\hat{p}_j(x_j(0))$ from any other nodes $j \neq l$ in $\mathcal{L}$ (see Problem 1). Line 3 is a condition to trigger the database update and the broadcast message design (see Lines 4 and 5, respectively). For Line 7, it calculates the local probability of $\bar{X}_l(t)$ based on the database $\mathcal{B}_l(t)$. Then, Line 8 gives the local estimated location $\hat{y}_l(t)$.

Above completes the description of the structure of the cooperative localization. Note that the parts to be design are Lines 4 and 5 in Algorithm 1.

C. DATABASE STRUCTURE

Recall that for each non-anchor node $l \in \mathcal{L}$, the database is updated after taking measurements, anchor states or messages (see Section V-B), and thus it just makes changes at time instants $t_{l,k}^{\mathrm{upd}} \in \mathcal{T}_l^{\mathrm{upd}} := \mathcal{T}_l^{\mathrm{ms}} \bigcup \mathcal{T}_l^{\mathrm{ac}} \bigcup \mathcal{T}_l^{\mathrm{rc}}$, where $\mathcal{T}_l^{\mathrm{ms}} := \bigcup_{j \in \mathcal{I}} \mathcal{T}_{l,j}^{\mathrm{ms}}$, $\mathcal{T}_l^{\mathrm{ac}} := \bigcup_{a \in \mathcal{A}} \mathcal{T}_{a,l}^{\mathrm{ms}}$, and $\mathcal{T}_l^{\mathrm{rc}} := \bigcup_{j \in \mathcal{L}} \mathcal{T}_{l,j}^{\mathrm{rc}}$ are the time sets of measurements, anchor measurements, and messages, respectively. At any given time instant t, there is the largest $t_{l,k}^{\mathrm{upd}} \in \mathcal{T}_l^{\mathrm{upd}}$ satisfying $t \geq t_{l,k}^{\mathrm{upd}}$. Then, the database $\mathcal{B}_l(t)$ $(l \in \mathcal{L})$ has the following form:

$$\mathcal{B}_l(t) = \mathcal{B}_l(t_{l,k}^{\mathrm{upd}}), \quad t_{l,k}^{\mathrm{upd}} = \max \mathcal{T}_l^{\mathrm{upd}} \bigcap [0,\ t],$$

which means the database keep unchanged before the next update time $t_{l,k+1}^{\mathrm{upd}}$. Thus, we just need to focus on $\mathcal{B}_l(t_{l,k}^{\mathrm{upd}}) =: \mathcal{B}_{l,k}$.

For $\mathcal{B}_{l,k}$, it contains the local priors and posteriors (see Definition 3), local measurement sets, and local anchor state sets over time window $[0,\ t_{l,k}^{\mathrm{upd}}]$. The detailed descriptions are given as follows.

For the local priors, they have the following form

$$\hat{p}_{l,k}^{\mathrm{a}}(t_{l,k,s}^{\mathrm{db,a}}) := \hat{p}_l(X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db,a}})|Z_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}], W_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}]), \tag{20}$$

where $t_{l,k,s}^{\mathrm{db,a}} \in \mathcal{T}_{l,k}^{\mathrm{db,a}}$. In (20), $\mathcal{L}_{X_{l,k}^{\mathrm{db,a}}(t_{l,k,s}^{\mathrm{db,a}})} \subseteq \mathcal{L}$ represents the local prior node set, and $Z_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}] \subseteq Z[t_{l,k,s}^{\mathrm{db,a}}]$ is the local prior measurement set, and $W_{l,k}^{\mathrm{db,a}}[t_{l,k,s}^{\mathrm{db,a}}] \subseteq Z[t_{l,k,s}^{\mathrm{db,a}}]$ is the local prior anchor state set.

Similarly, for the local posteriors, they have the following form

$$\hat{p}^{\mathrm{b}}_{l,k}(t^{\mathrm{db,b}}_{l,k,s}) := \hat{p}_l(X^{\mathrm{db,b}}_{l,k}(t^{\mathrm{db,b}}_{l,k,s})|Z^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db,b}}_{l,k,s}], W^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db,b}}_{l,k,s}]), \tag{21}$$

where $t^{\mathrm{db,b}}_{l,k,s} \in \mathcal{T}^{\mathrm{db,b}}_{l,k}$. In (21), $\mathcal{L}_{X^{\mathrm{db,b}}_{l,k}(t^{\mathrm{db,b}}_{l,k,s})} \subseteq \mathcal{L}$, $Z^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db,b}}_{l,k,s}] \subseteq Z[t^{\mathrm{upd}}_{l,k}]$, and $W^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db,b}}_{l,k,s}] \subseteq Z[t^{\mathrm{upd}}_{l,k}]$ are the local posterior node set, local posterior measurement set, and local posterior anchor state set, respectively.

For local measurement sets, they not only refer to those measurements taken by node l, but also include the measurements between other nodes which are obtained by received messages or anchor measurements. We label the measurement set in database $\mathcal{B}_{l,k}$ as

$$Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db,ms}}_{l,k,s}) \ni z_{i,j}(t^{\mathrm{db,ms}}_{l,k,s}), \quad t^{\mathrm{db,ms}}_{l,k,s} \in \mathcal{T}^{\mathrm{db,ms}}_{l,k}, \tag{22}$$

where $i, j \in \mathcal{I}$, and the local measurement set $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db,ms}}_{l,k,s})$ is a subset of the measurement set $Z(t^{\mathrm{db,ms}}_{l,k,s})$.

Similarly, for local anchor state sets, they not only refer to those only related to node l, but also contain the anchor states associated with other nodes. We label the anchor state set in database $\mathcal{B}_{l,k}$ as

$$W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db,ms}}_{l,k,s}) \ni x_a(t^{\mathrm{db,ms}}_{l,k,s}), \quad t^{\mathrm{db,ms}}_{l,k,s} \in \mathcal{T}^{\mathrm{db,ms}}_{l,k}. \tag{23}$$

For a clearer description of the database, we label

$$\mathcal{T}^{\mathrm{db}}_{l,k} = \mathcal{T}^{\mathrm{db,a}}_{l,k} \bigcup \mathcal{T}^{\mathrm{db,b}}_{l,k} \bigcup \mathcal{T}^{\mathrm{db,ms}}_{l,k} \tag{24}$$

as the local timeline in database $\mathcal{B}_{l,k}$. At each time $t^{\mathrm{db}}_{l,k,s} \in \mathcal{T}^{\mathrm{db}}_{l,k}$, we use a quadruple to contain the local prior, posterior, measurement set, and anchor state set, i.e.

$$b_{l,k}(t^{\mathrm{db}}_{l,k,s}) = \langle \hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}), \hat{p}^{\mathrm{b}}_{l,k}(t^{\mathrm{db}}_{l,k,s}), Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}), W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s})\rangle.$$

If at $t^{\mathrm{db}}_{l,k,s}$, there is no prior or posterior, then the prior or posterior is undefined. Likewise, if at $t^{\mathrm{db}}_{l,k,s}$, the measurement or anchor state set does not exist, then it is empty. To sum up, $\mathcal{B}_{l,k}$ has the following structure:

$$\mathcal{B}_{l,k} = \left\{ b_{l,k}(t^{\mathrm{db}}_{l,k,s}) : t^{\mathrm{db}}_{l,k,s} \in \mathcal{T}^{\mathrm{db}}_{l,k} \right\}.$$

D. BROADCAST MESSAGE

In the distributed prior-cut algorithm, only local priors, measurement sets, and anchor state sets are transmitted. Thus, the broadcast messages do not include any local posterior from the database. To be more specific, each broadcast message has the following form

$$m^{\mathrm{bc}}_{l,k} = \left\{ b^{\mathrm{bc}}_{l,k}(t) : t \in \bar{\mathcal{T}}^{\mathrm{db}}_{l,k} \right\},$$

where $b^{\mathrm{bc}}_{l,k}(t) = \langle \hat{p}^{\mathrm{a}}_{l,k}(t), Z^{\mathrm{db,ms}}_{l,k}(t), W^{\mathrm{db,ms}}_{l,k}(t)\rangle$. Also, $\bar{\mathcal{T}}^{\mathrm{db}}_{l,k} \subseteq \mathcal{T}^{\mathrm{db}}_{l,k}$ such that $b_{l,k}(t^{\mathrm{db}}_{l,k,s}) \neq b_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$. That means each $m^{\mathrm{bc}}_{l,k}$ only contains the information different from that in the previous database.

E. RECEIVED MESSAGE

Since not all the parts of a broadcast message can be successfully received, the received message is a part of the broadcast message. Recall that at $t^{\mathrm{rc}}_{l,j,k}$, node l receives message $m^{\mathrm{rc}}_{l,j}(t^{\mathrm{rc}}_{l,j,k})$ from node j (see Section II). To unify the time index, we still consider the time index $t^{\mathrm{upd}}_{l,k}$, since $\mathcal{T}^{\mathrm{rc}}_l \subseteq \mathcal{T}^{\mathrm{upd}}_l$ (see Section V-C). At $t^{\mathrm{upd}}_{l,k}$, if node l receives message from node j, then we label it as

$$m^{\mathrm{rc}}_{l,j}(t^{\mathrm{upd}}_{l,k}) = \{\mu^{\mathrm{rc}}_{l,j}(t^{\mathrm{mg}}_{l,j,k,s}) : t^{\mathrm{mg}}_{l,j,k,s} \in \mathcal{T}^{\mathrm{mg}}_{l,j,k}\},$$

where $\mu^{\mathrm{rc}}_{l,j}(t^{\mathrm{mg}}_{l,j,k,s})$ is a triple containing the local prior, measurement set, and anchor state set from non-anchor node j, i.e.

$$\mu^{\mathrm{rc}}_{l,j}(t^{\mathrm{mg}}_{l,j,k,s}) = \langle \hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s}), Z^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s}), W^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})\rangle,$$

where

$$\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s}) = \hat{p}_j(X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})|Z^{\mathrm{db,a}}_{j,k'}[t^{\mathrm{mg}}_{l,j,k,s}], W^{\mathrm{db,a}}_{j,k'}[t^{\mathrm{mg}}_{l,j,k,s}])$$

and the local prior, measurement set, and anchor state set correspond to those in (20), (22), and (23), respectively.

We define $M^{\mathrm{rc}}_{l,j}(t^{\mathrm{upd}}_{l,k}) := \{m^{\mathrm{rc}}_{l,j}(t^{\mathrm{upd}}_{l,k})\}$, and let $M^{\mathrm{rc}}_{i}(t^{\mathrm{upd}}_{l,k}) := \bigcup_{j\in\mathcal{I}} M^{\mathrm{rc}}_{l,j}(t^{\mathrm{upd}}_{l,k})$, and also set $\mathcal{T}^{\mathrm{mg}}_{l,k} = \bigcup_{j\in\mathcal{I}} \mathcal{T}^{\mathrm{mg}}_{l,j,k}$. Then, it is readily to design the database update in Section V-F.

F. DATABASE UPDATE

This section gives the key methods in the distributed prior-cut algorithm. The main idea is to cut the local prior at each node and at each time instant into two parts such that the estimation gap (measured by MSE) is reduced when the irrelevant part (to the location estimation) is discarded. The details are given as follows.

At each time instant $t^{\mathrm{upd}}_{l,k}$, the database of each non-anchor node $l \in \mathcal{L}$ is updated from $\mathcal{B}_{l,k-1}$ to $\mathcal{B}_{l,k}$ by the newly arrived measurements, anchor states, or messages, which is $Z_l(t^{\mathrm{upd}}_{l,k}) := \bigcup_{j\in\mathcal{I}} Z_{l,j}(t^{\mathrm{upd}}_{l,k})$, $W_l(t^{\mathrm{upd}}_{l,k}) := \bigcup_{a\in\mathcal{A}} W_{l,a}(t^{\mathrm{upd}}_{l,k})$ or $M^{\mathrm{rc}}_i(t^{\mathrm{upd}}_{l,k})$, respectively. The database update refreshes the local timeline [see (24)], measurement set, anchor state set, prior, and posterior, respectively.

We provide the local timeline update first, since it is the foundation of the other updates. Due to the limited computational capability and the storage space, $\mathcal{B}_{l,k}$ cannot include every information in the past, especially when k goes large. We use $N^{\max}_l$, the largest number of time instants to constrain the size of $\mathcal{B}_{l,k}$, which means the total number of time instants included in the database cannot exceed this number. As a result, the updated timeline $\mathcal{T}^{\mathrm{db}}_{l,k}$ [see (24)] contains $N^{\max}_l$ most recent time instants in time set $\mathcal{T}^{\mathrm{all}}_{l,k}$ defined as follows

$$\mathcal{T}^{\mathrm{all}}_{l,k} := \begin{cases} \mathcal{T}^{\mathrm{db}}_{l,k-1} \bigcup \{t^{\mathrm{upd}}_{l,k}\} \bigcup \bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}, & \\ & \text{if } Z_l(t^{\mathrm{upd}}_{l,k}) \neq \emptyset \text{ or } W_l(t^{\mathrm{upd}}_{l,k}) \neq \emptyset, \\ \mathcal{T}^{\mathrm{db}}_{l,k-1} \bigcup \bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}, & \text{otherwise}, \end{cases}$$

where $\bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}$ is the subset of $\mathcal{T}^{\mathrm{mg}}_{l,k}$ satisfying at least one of the following three conditions:

1) $\forall t^{\mathrm{mg}}_{l,j,k,s} \in \bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}$, $\mathcal{L}_{X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})} \bigcap \mathcal{L}_{X^{\mathrm{db,a}}_{l,k}(\max \mathcal{T}^{\mathrm{db}}_{l,k-1})} \neq \emptyset$;
2) $\forall t^{\mathrm{mg}}_{l,j,k,s} \in \bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}$, $\mathcal{V}(Z^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})) \bigcap \mathcal{L}_{X^{\mathrm{db,a}}_{l,k}(\max \mathcal{T}^{\mathrm{db}}_{l,k-1})} \neq \emptyset$;
3) $\#\mathcal{L}_{X^{\mathrm{db,a}}_{l,k}(\max \mathcal{T}^{\mathrm{db}}_{l,k-1})} \leq L^{\max}_{l}$.

Condition 1) tells that for each time instant $t^{\mathrm{mg}}_{l,j,k,s}$ in $\bar{\mathcal{T}}^{\mathrm{mg}}_{l,k}$, the corresponding local node set $\mathcal{L}_{X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})}$ in $\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{mg}}_{l,j,k,s})$ from a received message must has a non-empty intersection with $\mathcal{L}_{X^{\mathrm{db,a}}_{l,k}(\max \mathcal{T}^{\mathrm{db}}_{l,k-1})}$ in node l's database. It implies that we should utilize those messages whose local prior is related to the most recent local prior in the previous database. Similarly, Condition 2) means we should consider those messages whose measurement set is correlated to the most recent local prior in the previous database. Condition 3) says that if the number of nodes stored in the previous database is smaller than a level $L^{\max}_{l}$,[11] the received message should be unconditionally included in the database update. Above complete the description of timeline update. For the other updates, they are given in Algorithm 2.

Algorithm 2 Distributed Prior-Cut Algorithm: Database Update

1: **for** $t^{\mathrm{db}}_{l,k,s} := \min \mathcal{T}^{\mathrm{db}}_{l,k}$ **to** $\max \mathcal{T}^{\mathrm{db}}_{l,k}$ **do**
2:   **if** $t^{\mathrm{db}}_{l,k,s} == t^{\mathrm{upd}}_{l,k}$ **then**
3:     $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = Z_l(t^{\mathrm{upd}}_{l,k}) \bigcup Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
4:     $W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = W_l(t^{\mathrm{upd}}_{l,k}) \bigcup W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
5:   **else**
6:     $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) \bigcup Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
7:     $W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) = W^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) \bigcup W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$;
8:   **end if**
9:   **for each** valid l' **do**
10:     $j^{*}_{l,l',k,s} = \arg\min_{j} \left\{ \widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right] : j \in \mathcal{J}^{\mathrm{rc}}_{l,k,s} \bigcup \{l,-1\} \right\}$;
11:   **end for**
12:   $\tilde{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) = \prod_{j \in \mathcal{J}^{*}_{l,k,s}} \hat{p}^{\mathrm{a}}_{j,k'}\left(t^{\mathrm{db}}_{l,k,s}, \mathcal{J}^{*}_{l,k,s}(j)\right)$;
13:   $\langle \bar{Z}^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}), X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s})\rangle = \arg\min \Delta^{\mathrm{db}}_{l,k,s}$;
14:   $\hat{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \leftarrow (26)$;
15:   **if** $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \neq \emptyset || W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) \neq \emptyset$ **then**
16:     $\hat{p}_l(X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s}) | Z^{\mathrm{db,b}}_{l,k}[t^{\mathrm{db}}_{l,k,s}]) \leftarrow (27)$;
17:   **end if**
18:   **if** $t^{\mathrm{db}}_{l,k,s} < \max \mathcal{T}^{\mathrm{db}}_{l,k}$ **then**
19:     $\hat{p}_{-1}(X^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db}}_{l,k,s+1}) | Z^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db}}_{l,k,s}]) \leftarrow (28)$;
20:   **end if**
21: **end for**

In Algorithm 2, the database $\mathcal{B}_{l,k}$ is updated element by element from $t^{\mathrm{db}}_{l,k,s} = \min \mathcal{T}^{\mathrm{db}}_{l,k}$ to $\max \mathcal{T}^{\mathrm{db}}_{l,k}$. Lines between 1 and 21 provide the detailed update of $b_{l,k}(t^{\mathrm{db}}_{l,k,s})$, which contain five building blocks: local measurement and anchor state update, prior fusion, prior cut, posterior update, and prior prediction. Note that the database for each non-anchor node $l \in \mathcal{L}$ is initialized as

$$\mathcal{B}_{l,0} := \mathcal{B}_l(0) = \left\{ \langle \hat{p}_l(x_l(0)), \uparrow, \uparrow, \uparrow \rangle \right\}, \tag{25}$$

which further specifies the initial condition in Algorithm 1. In (25), each $\uparrow$ stands for an undefined term. For $\mathcal{B}_{l,0}$, the local posterior, measurement set, and anchor state set are undefined.

1) LOCAL MEASUREMENT AND ANCHOR STATE UPDATE

From Line 2 to Line 8 of Algorithm 2, the local measurement set $Z^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s})$ and the local anchor state set $W^{\mathrm{db,ms}}_{l,k}(t^{\mathrm{db}}_{l,k,s})$ in $b_{l,k}(t^{\mathrm{db}}_{l,k,s})$ are updated. If $t^{\mathrm{db}}_{l,k,s}$ equals $t^{\mathrm{upd}}_{l,k}$, then the updated local measurement set is composed by the set of newly taken measurements $Z_l(t^{\mathrm{upd}}_{l,k})$ and the set of measurements from the old database $Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (see Line 3); and the updated local anchor state set consists of newly received local anchor state set $W_l(t^{\mathrm{upd}}_{l,k})$ and the local anchor state set in the old database $W^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (see Line 4). If $t^{\mathrm{db}}_{l,k,s}$ is not equal to $t^{\mathrm{upd}}_{l,k}$, then the updated measurement set includes the set of measurements provided by the newly arrived messages $Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s})$ and the set of measurements from the old database $Z^{\mathrm{db,ms}}_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (see Line 6), where $Z^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) := \bigcup_{j\in\mathcal{I}} Z^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,k,s})$; and similarly the updated local anchor state set consists of the local anchor state sets provided by the newly arrived messages and the old database, respectively (see Line 7), where $W^{\mathrm{mg,ms}}_{l,k}(t^{\mathrm{mg}}_{l,k,s}) := \bigcup_{j\in\mathcal{I}} W^{\mathrm{mg,ms}}_{j,k'}(t^{\mathrm{mg}}_{l,k,s})$ in Line 7.

2) LOCAL PRIOR FUSION

Lines 9-12 of Algorithm 2 return the fused local prior

$$\tilde{p}^{\mathrm{a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) := \hat{p}_l(\tilde{X}^{\mathrm{db,a}}_{l,k}(t^{\mathrm{db,a}}_{l,k,s}) | Z^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db,a}}_{l,k,s}], W^{\mathrm{db,a}}_{l,k}[t^{\mathrm{db,a}}_{l,k,s}]),$$

where $t^{\mathrm{db,a}}_{l,k,s} \in \mathcal{T}^{\mathrm{db,a}}_{l,k}$. Firstly, we define the local prior variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right]$ as follows:

$$\int_{\mathcal{X}^{\mathrm{db,a}}_{j,k',s}} \left\{ \widehat{\mathbb{E}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right] - x_{l'}(t^{\mathrm{db}}_{l,k,s}) \right\}^2 \times \hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}) \mathrm{d}X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}),$$

where $k' = k$ for $j = l$, $X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}) \in \mathcal{X}^{\mathrm{db,a}}_{j,k',s}$, and

$$\widehat{\mathbb{E}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right] := \int_{\mathcal{X}^{\mathrm{db,a}}_{j,k',s}} x_{l'}(t^{\mathrm{db}}_{l,k,s}) \hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}) \mathrm{d}X^{\mathrm{db,a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s}).$$

Variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right]$ reflects the uncertainty of $x^{\mathrm{a}}_{l'}(t^{\mathrm{db}}_{l,k,s})$ under local prior $\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})$ which comes from: $b_{l,k-1}(t^{\mathrm{db}}_{l,k,s})$ (i.e. $j = l$), $\mu^{\mathrm{rc}}_{l,j}(t^{\mathrm{db}}_{l,k,s})$ (i.e. $j \notin \{l, -1\}$, and all such j form the set $\mathcal{J}^{\mathrm{rc}}_{l,k,s}$) or the predicted local prior from the previous time instant (i.e. $j = -1$, see Section V-F5). For different j but the same state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$, the variance $\widehat{\mathrm{Var}}^{\mathrm{a}}_{j}\left[x_{l'}(t^{\mathrm{db}}_{l,k,s})\right]$ differs. The smaller the variance is, the less uncertainty the state $x_{l'}(t^{\mathrm{db}}_{l,k,s})$ should have. Thus, the marginal pdf of $\hat{p}^{\mathrm{a}}_{j,k'}(t^{\mathrm{db}}_{l,k,s})$ w.r.t. state

[11] It is a parameter in this algorithm which constrains the sizes of prior and posterior, see Section V-F3.

$x_{l'}(t^{\text{db}}_{l,k,s})$ with the smallest variance should be chosen as the updated marginal pdf. This process is conducted from Line 9 to Line 11 for each valid state $x_{l'}(t^{\text{db}}_{l,k,s})$, i.e. it should be included in at least one of the message $m^{\text{rc}}_{l,j}(t^{\text{upd}}_{l,k})$, or in the old database $\mathcal{B}_{i,k-1}$, or in the predicted prior from $t^{\text{db}}_{i,k,s-1}$. Note that if $s = 1$, the predicted pdf is derived from the nearest time instant in the old database. For each valid l', the best marginal prior's index[12] $j^*_{l,l',k,s}$ is selected by Line 10 among indices $j \in \mathcal{J}^{\text{rc}}_{l,k,s} \bigcup \{l, -1\}$. All such $j^*_{l,l',k,s}$ (for all valid l') form the set $\mathcal{J}^*_{l,k,s}$, and we set $\mathcal{J}^*_{l,k,s}(j) = \{j' : j^*_{l,l',k,s} = j\}$, and

$$\hat{p}^{\text{a}}_{j,k'}\left(t^{\text{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right) = \int_{\check{\mathcal{X}}^*_{j,k,s}} \hat{p}^{\text{a}}_{j,k'}\left(t^{\text{db}}_{l,k,s}\right) \mathrm{d}\check{X}^*_{j,k,s},$$

where $\mathcal{L}_{\check{X}^*_{j,k,s}} = \mathcal{L}_{X^{\text{db,a}}_{j,k'}(t^{\text{db}}_{l,k,s})} \setminus \mathcal{J}^*_{l,k,s}(j)$, and $\check{X}^*_{j,k,s} \in \check{\mathcal{X}}^*_{j,k,s}$. Then, we reconstruct the local prior (i.e. the fused local prior) from the selected marginal priors, which is given in Line 12: For the marginal priors from node j, they still remain the same joint distribution $\hat{p}^{\text{a}}_{j,k'}\left(t^{\text{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right)$. Since the relationship among the priors in different nodes j are unknown, we assume they are independent of each other, which means $\tilde{p}^{\text{a}}_{l,k}(t^{\text{db,a}}_{l,k,s})$ is the product of different $\hat{p}^{\text{a}}_{j,k'}\left(t^{\text{db}}_{l,k,s}, \mathcal{J}^*_{l,k,s}(j)\right)$.

Remark 7: Note that the marginal priors can be from all possible nodes who broadcast messages in this network, and as a result, the fused local prior would contain a large number of nodes' information. Thus, it is impractical if

- *we put the fused local prior in the broadcast message, since the communication cost would be very large;*
- *the whole fused local prior is used to do further calculations (including Bayesian inference and local prior prediction), because the computational capability of a node is limited;*
- *we store the fused local prior in the new database, as the storage space is limited.*

Therefore, it is necessary to reduce the size of the fused local prior. This size reduction is based on the local prior cut (see Section IV-B) and given in Lines 13 and 14 of Algorithm 2.

3) LOCAL PRIOR CUT

We explain the meaning of Line 13 as follows. Firstly, we define the state-number-constraint AMS (SNCAMS) triple. Similar to the AMS triple in Definition 2, we need the local measurement graph based on the local measurement set $Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})$, i.e.

$$\mathcal{G}^{\text{db}}_{l,k,s} := \left(\mathcal{V}(Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})), \mathcal{E}(Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s}))\right).$$

Then the local measurement cluster can be defined as $\mathcal{C}^{\text{db}}_{l,k,s,r}$, where $r \in \{1, \ldots, R^{\text{db}}_{l,k,s}\} =: \mathcal{R}^{\text{db}}_{l,k,s}$. Thus, the (local) AMS triple $\langle \bar{Z}^{\text{db,ms}}_{i,k}(t^{\text{db}}_{l,k,s}), X^{\text{db,a}}_{l,k}(t), \bar{W}^{\text{db,ms}}_{i,k}(t^{\text{db}}_{l,k,s})\rangle$ is defined according to Definition 2, where $\bar{Z}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s}) \subseteq Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})$, $X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}) \subseteq \check{X}^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s})$, and $\bar{W}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s}) \subseteq W^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})$. To give the definition of SNCAMS triple, we introduce the state-generating node set $\mathcal{L}^{\text{gen}}_{l,k,s}$, and it has the following relationship with $X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s})$:

$$\mathcal{L}_{X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s})} = \Big\{l' : (l', j) \in \mathcal{E}(Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})) \vee (j, l') \in \mathcal{E}(Z^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})), j \in \mathcal{L}^{\text{gen}}_{l,k,s}\Big\}.$$

If we only consider the state-generating node set with number constraint $L^{\max}_l$, i.e. $\#\mathcal{L}^{\text{gen}}_{l,k,s} = L^{\max}_l$, then the generated AMS triple is called $L^{\max}_l$-SNCAMS triple.

For $L^{\max}_l$-SNCAMS triples, we have the similar results as those in Theorem 1: Cut local prior $\tilde{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})$ into two sub-priors, i.e. $\hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})$ and $\tilde{p}^{\text{left}}_{l,k,s}$ which are the local priors of $X^{\text{db,a}}_{l,k}(t)$ and $\check{X}^{\text{db,a}}_{l,k}(t)$, respectively. From the perspective of node l, if the cut gap is small enough, then there is no big difference between the original local location estimation and the estimation only considering the $L^{\max}_l$-SNCAMS triple.

Let $\mathcal{U}^{\text{db}}_{l,k,s}(L^{\max}_l)$ be the set of all possible $L^{\max}_l$-SNCAMS triples at $t = t^{\text{db}}_{l,k,s}$, which means it is the set of $\langle \bar{Z}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s}), X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}), \bar{W}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})\rangle$ with states-number constraint $L^{\max}_l$, labeled as $u^{\text{db}}_{l,k,s}$. Then, we define the set of cut gaps corresponding to $\mathcal{U}^{\text{db}}_{l,k,s}(L^{\max}_l)$ as follows

$$\Delta^{\text{db}}_{l,k,s} = \left\{ \left\|\tilde{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s}) - \hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})\tilde{p}^{\text{left}}_{l,k,s}\right\|_\infty : u^{\text{db}}_{l,k,s} \in \mathcal{U}^{\text{db}}_{l,k,s}(L^{\max}_l)\right\}.$$

Line 13 selects the SNCAMS triple corresponding to the smallest element in $\Delta^{\text{db}}_{l,k,s}$, and finally Line 14 gives the local prior $\hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})$ stored in the database, specifically:

$$\hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s}) = \int_{\check{\mathcal{X}}^{\text{db,a}}_{l,k,s}} \tilde{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s}) \mathrm{d}\check{X}^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}), \tag{26}$$

where $\check{X}^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}) \in \check{\mathcal{X}}^{\text{db,a}}_{l,k,s}$.

4) LOCAL POSTERIOR UPDATE

Lines 15-17 provide the local posterior update. If the local measurement set or anchor state set is not empty, then the local posterior can be derived by Bayes' rule

$$\hat{p}^{\text{a}}_{l,k}(t^{\text{db,b}}_{l,k,s}) = \frac{p(\bar{Z}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})|X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}))\hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})}{\text{Normalization}}. \tag{27}$$

where

$$\text{Normalization} = \int_{\mathcal{X}^{\text{db,a}}_{l,k,s}} p(\bar{Z}^{\text{db,ms}}_{l,k}(t^{\text{db}}_{l,k,s})|X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s})) \times \hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})\mathrm{d}X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}).$$

5) LOCAL PRIOR PREDICTION

From Line 18 to Line 20, the updated local posterior is employed to predict the local prior at the next time instant $t^{\text{db}}_{l,k,s+1}$, if $t^{\text{db}}_{l,k,s}$ is not the last time instant in $\mathcal{T}^{\text{db}}_{l,k}$. This prediction is given in Line 19, where

$$\hat{p}^{\text{a}}_{-1,k}(t^{\text{db}}_{l,k,s+1}) = \int_{\mathcal{X}^{\text{db,a}}_{l,k,s}} p(X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s+1})|X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s})) \times\hat{p}^{\text{a}}_{l,k}(t^{\text{db}}_{l,k,s})\mathrm{d}X^{\text{db,a}}_{l,k}(t^{\text{db}}_{l,k,s}). \tag{28}$$

[12]This means the best marginal prior is provided by node $j^*_{l,l',k,s}$.

G. ALGORITHM SUMMARY

We give a summary of the distributed prior-cut algorithm in this subsection. For each non-anchor node $l \in \mathcal{L}$, it has an initial prior $\hat{p}_l(x(0))$ on its state $x_l(0)$ at $t = 0$. Since node l is not able to obtain its location $y_l(t)$ (see Section II), it takes measurements from other nodes (see Section II-A), and broadcasts/receives messages to/from other nodes (see Section II-B). Note that node l does not know when the next measurement or the next message will come (see Section III-A). Based on the measurements and messages, the distributed prior-cut algorithm estimates $y_l(t)$ in a real-time manner (the structure of this algorithm is given in Algorithm 1):

- If node l takes a measurement from a non-anchor node, then the database (whose structure is shown in Section V-C) is updated according to this measurement.
- If node l takes a measurement from an anchor node, then it will receive the anchor state from that anchor node immediately (see Section III-A1), and the database is updated according to this measurement and the anchor state.
- If an anchor node takes a measurement from node l, then it will receive this measurement as well as the anchor state immediately (see Section III-A1), and the database is updated according to this measurement and the anchor state.
- If node l receives a message (the structure of the received message is given in Section V-E), then the database is updated according to this message.

Note that these four situations can happen simultaneously. The detailed database update, including the design of local probability, is given in Section V-F. After updating the database, node l designs the broadcast message accordingly (see Section V-D). We stress that the designed message is not necessarily to be broadcasted directly due to the limited bandwidth, and whether it can be successfully broadcasted is dependent on the MAC layer (an example is given in Appendix C which is employed by our simulation examples in Section VI). After designing the broadcast message, node l calculates the local prior based on the updated database, and estimates its location $y_l(t)$ correspondingly. Note that even if no measurements or messages arrive, node l still needs to calculate the local prior and estimate its location $y_l(t)$ but based on the current database (without update). Measurements and messages play the role in updating the database and designing broadcast message.

VI. SIMULATION RESULTS

In this section, simulations are given to corroborate the effectiveness of our proposed distributed prior-cut algorithm. We consider two different localization problems: the mobile user cooperative localization (Section VI-A), and the UAV localization in scanning task (Section VI-B). Note that the simulations are conducted under practical wireless network settings, where the physical and MAC layers are properly modeled.

A. MOBILE USER COOPERATIVE LOCALIZATION

Consider a network with 25 mobile users ($I = 25$), where only 5 users (anchor nodes) can access their real-time locations timely. For example, these locations can be obtained from the GPS/DGPS signals, base station measurements for outdoor positioning systems, or WiFi based localization for indoor positioning systems. For the other 20 users (non-anchor nodes), they conduct the self-localization cooperatively by using our proposed distributed prior-cut algorithm during time duration $[0,\ 50]$.

1) MOBILITY MODEL

The initial locations of these 25 users are uniformly distributed in a 60m $\times$ 60m area, see Fig. 6(a) for one realization. Each node does not know the initial location accurately, but we assume its guess on the initial location is within a 10m $\times$ 10m area centered at the true location. The mobility model of mobile users is govern by a 2-dimensional Browian motion, which can be described by (1) and (2) with

$$f_i(x_i(t), t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad g_i(x_i(t)) = x_i(t), \quad Q_i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

where $x_i(t), y_i(t) \in \mathbb{R}^2$ and $i \in \mathcal{I}$. One realization of the location trajectories is given in Fig. 6(b), which indicates how the node locations change over time (the initial locations are given in Fig. 6(a)).

2) MEASUREMENT MODEL

Each node can take measurements from at most 5 neighbors at one time instant, and we assume these 5 neighbors are nearest to the node.[13] For each measurement link, the measuring time instant follows the Poisson process with rate $\lambda = 5$, i.e. a measurement is taken every 0.2 second on average. We assume the measurement noise in (3) follows a zero-mean Gaussian distribution with standard deviation 0.1, i.e.

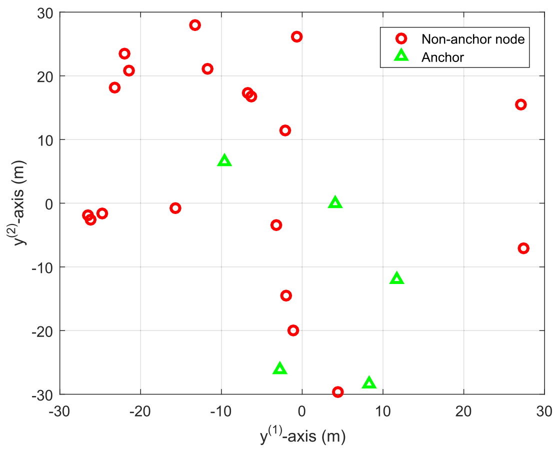
$$v_{i,j}(t^{\text{ms}}_{i,j,k}) \sim \text{N}(0, 0.1),$$

which means the measurement accuracy is at the level of 0.1m. Note that the neighbors are changing over time due to the mobility of users.

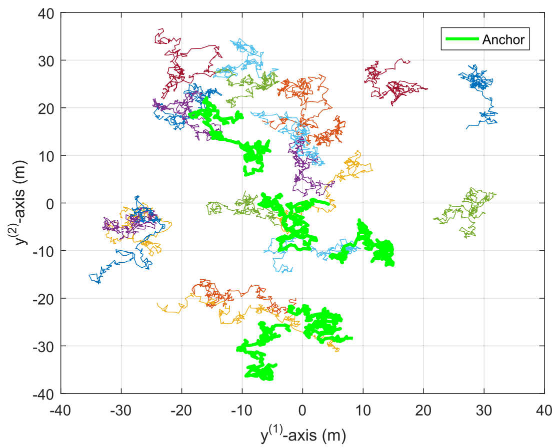
3) COMMUNICATION MODEL

In the distributed prior-cut algorithms, the non-anchor nodes interchange their information through broadcasting in wireless communications. We assume all the non-anchor nodes share a $B = 40$MHz communication bandwidth. To present our simulation in a more explicit way, we move the communication details in Appendix C, where the physical layer power control is with the on-off structure and the media access control (MAC) layer protocol is carrier-sense multiple access (CSMA). It should be noted that the communications

[13] It should be noted that the node never knows its distances between their neighbors. This assumption just implies that the measurement can be taken more easily between closeby nodes, and the other measurements corresponding to longer distances are neglected in the simulation.



(a)



(b)

FIGURE 6. An example of the mobile users' motions: (a) initial locations, where the circles and triangles are the locations of non-anchor nodes and anchor nodes, respectively; (b) location trajectories, where the thin and thick lines represent the trajectories of non-anchor nodes and anchor nodes, respectively.

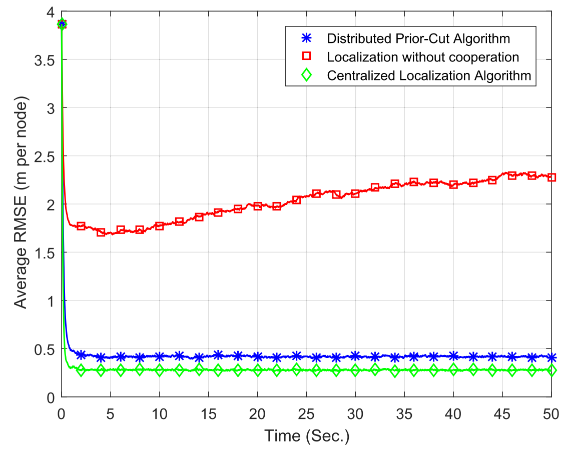


FIGURE 7. Comparisons of the distributed prior-cut algorithm, the localization without cooperation, and the centralized localization algorithm for mobile user cooperative localization, averaged over 100 simulation runs with different initial node locations. The average root-mean-square error (RMSE) means the square root of $\mathbb{E}\|\hat{y}_l(t) - y_l(t)\|$ (see Problem 1) averaged by all nodes $l \in \mathcal{L}$. The average RMSE of the distributed prior-cut algorithm converges to 0.43m/node within 2 seconds, which is comparable to the centralized algorithm. For the localization without cooperation, even though the average RMSE experienced a decrease (around 1.7m/node) during time interval [0, 5], it goes worse and reach 2.3m/node at $t = 50$.

are fully asynchronous, i.e. they are not scheduled to happen exactly in each time slot synchronized by the whole network. Also, there is no delay for processing the broadcast, but the broadcast process takes time since the data rate is limited (more details can be found in Appendix C).

4) DISTRIBUTED PRIOR-CUT ALGORITHM

Our proposed algorithm (the details are given in Section V) only has two parameters: one is the memory length $N_l^{\max}$, and the other is the constraint for state-generating node set $L_l^{\max}$ which determines the size of local prior. In this problem, we set $N_l^{\max} = 5$ and $L_l^{\max} = 8$. For the prior prediction and posterior update, we use the continuous-discrete unscented Kalman filter (UKF).[14]

[14] Since the local priors and posteriors are approximated by Gaussian distribution in the UKF, the cut gap is not easy to calculate (for particle filters, the cut gap is easy to calculate), and for the local prior cut in Section V-F3, we use the linear correlation (averaged on each component of a state) instead. This is reasonable, because the correlation reflects the dependence of two Gaussian variables.

5) RESULTS AND COMPARISONS

We compare the average RMSE among the distributed prior-cut algorithm, the localization without cooperation, and the centralized localization algorithm for 100 simulation runs. The results are shown in Fig. 7. We can see that on average, each node's localization error is within 0.5m. This result is very close to the centralized algorithm and much better than the localization algorithm without cooperation.

B. UAV LOCALIZATION IN SCANNING TASK

Consider 45 UAVs carrying out a scanning task for a region by hovering around the same point, where only 5 UAVs (anchor nodes) can access their own location timely and the other 40 UAVs are non-anchor nodes to be localized. The simulation period is [0, 50].

1) MOBILITY MODEL

The initial locations of these 45 UAVs are uniformly placed in a 500m × 500m × 500m region [100, 600] × [100, 600] × [300, 800]. Similar to Section VI-A1, each UAV does not know the initial location, but we assume its guess on the initial location is within a 100m × 100m × 100m region centered at the true location. The mobility model of UAVs is govern by (1) and (2) with

$$f_i(x_i(t), t) = \begin{bmatrix} 0 & -\omega_i & 0 \\ \omega_i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x_i(t), \quad g_i(x_i(t)) = x_i(t),$$

$$Q_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad i \in \mathcal{I}, \tag{29}$$

where $x_i(t)$, $y_i(t) \in \mathbb{R}^3$. In (29), ω_i is the hover angular velocity for UAV i. For this problem, we assume $\omega_1 = \omega_3 = \ldots = \omega_{45} = 0.1\text{rad/s}$ and $\omega_2 = \omega_4 = \ldots = \omega_{44} = -0.1\text{rad/s}$, in which the positive angular velocity means the hover is counter-clock-wise from the bird's-eye view. One realization of the location trajectories is given in Fig. 8, where two non-anchor nodes and an anchor node are highlighted.

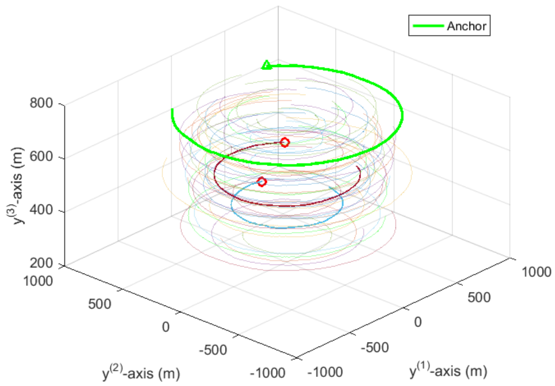


FIGURE 8. **Location trajectories of UAVs. From the top to bottom, three highlighted UAVs are UAV 42 (non-anchor nodes), UAVs 7 and 3 (anchor nodes), respectively. For these three UAVs, the triangle and circles label the initial locations of anchor and non-anchor nodes, respectively.**

2) MEASUREMENT AND COMMUNICATION MODELS

The measurement model is the same as that in Section VI-A2. For communication model, we still use a similar model in Section VI-A3 (these parameters below can be found in Appendix C) with a small difference highlighted as follows: The transmit power is 0.5W. The path loss exponent is 2. The silence distance is 20m.

3) RESULTS AND COMPARISONS

For the distributed prior-cut algorithm, we use the same parameters as that in Section VI-A4, i.e. $N_l^{\max} = 5$ and $L_l^{\max} = 8$. For the prior prediction and posterior update, we still use the UKF. The comparison of the distributed prior-cut algorithm, the localization without cooperation, and the centralized algorithm averaged by 100 simulation runs is given in Fig. 9. We can see that the distributed prior-cut algorithm has a close performance to the centralized algorithm, and outperforms the localization without cooperation.

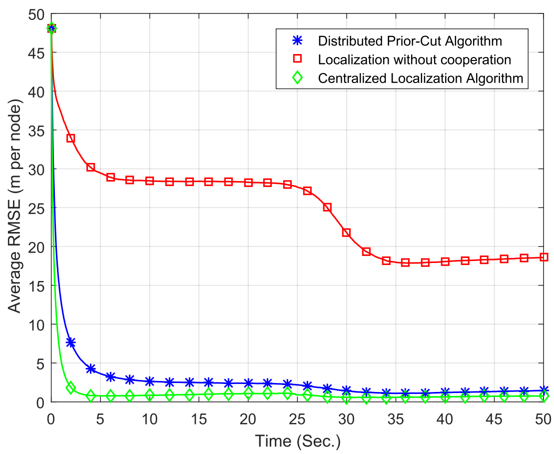


FIGURE 9. **Comparisons for UAV localization in scanning task, averaged over 100 simulation runs with different initial node locations. The average RMSE of the distributed prior-cut algorithm reaches 3.5m/node at $t = 5$ and 1.5m/node at $t = 30$, which is comparable to the centralized algorithm. For the localization without cooperation, the average RMSE cannot be smaller than 17m/node.**

VII. CONCLUSION

In this paper, the mobile nodes' cooperative localization problem with asynchronous communications and measurements has been studied. We have modeled the node mobility by using the stochastic differential equation, and employed the continuous-discrete Bayesian filter to give the centralized localization algorithm which utilizes the global information. Important concepts including Admissible Measurement-State (AMS) triple, prior cut, and cut gap, have been introduced in analyzing the estimation gap between the centralized algorithm and the algorithm with an AMS triple. We have proved that if the cut gap is small enough, i.e. the cut two parts are nearly independent, then the estimation gap can also be very small. With this important property, we have designed the distributed prior-cut algorithm to solve the cooperative localization problem with asynchronous communications and measurements.

The presented work serves as the first step to develop the asynchronous localization problems. For future work, it is meaningful to consider the outlier and structural noises [25] for practical applications which leads to a noise-tolerant cooperative localization. We are also considering to implement our proposed prior-cut algorithm to multi-UAV systems. To improve the estimation accuracy, we will place multiple nodes in one UAV, which is inspired by [26] (see Fig. 10). Note that this design can also enable the attitude estimation of each UAV.

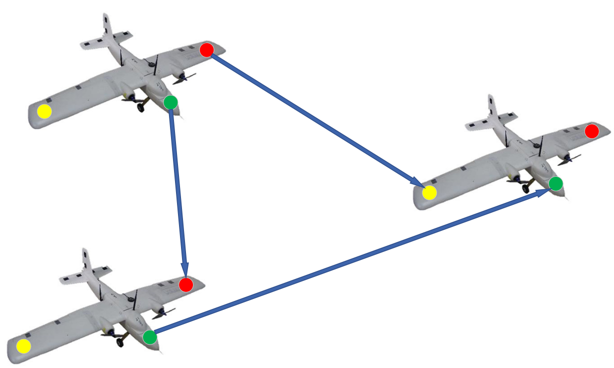


FIGURE 10. **Multiple (three) nodes assembled in each UAV.**

Another line of future work can consider the cooperative localization in an $\mathcal{H}_\infty$ manner. Different from the Bayesian filtering framework utilized in the current work which relies on the exact probability density function of the noises, the $\mathcal{H}_\infty$ filter does not need any knowledge on the probability measure (see [27], [28]). Therefore, it will be more practical to use $\mathcal{H}_\infty$ as the performance measure in cooperative localization problems.

It is also very meaningful to analyze important properties of the distributed prior-cut algorithm, e.g., the stability (or convergence) and the convergence rate. Since the overall stability and convergence rate are difficult to prove for nonlinear stochastic systems, we would simplify the system model to a linear system and then conduct the analyses accordingly.

APPENDIX A
PROOF OF LEMMA 1

This proof is divided into two parts: In the first part, we prove the recursive equations [i.e. (8) in the prediction step and (9) in the update step] hold. As a result, $p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}])$ can be derived sequentially from $t=0$. In the second part, we show that equation (7) holds.

i) In the prediction step, $p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])$ can be written as

$$p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) = \int_{\mathcal{X}} p(X(t_k^{\rm ms}), X(t_{k-1}^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]){\rm d}X(t_{k-1}^{\rm ms}), \quad (30)$$

because the marginal distribution can be obtained by integration. Then, in equation (30), we write $p(X(t_k^{\rm ms}), X(t_{k-1}^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])$ as

$$\begin{aligned} &p(X(t_k^{\rm ms})|X(t_{k-1}^{\rm ms}), Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) \\ &\quad\times p(X(t_{k-1}^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) \\ &\overset{(a)}{=} p(X(t_k^{\rm ms})|X(t_{k-1}^{\rm ms}))p(X(t_{k-1}^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]), \end{aligned} \quad (31)$$

where (a) holds with the Markov property of sequence $\{X(t_k^{\rm ms})\}_{k\in\mathcal{K}}$. Combining (30) and (31), we have (8).

In the update step, we rewrite the posterior

$$p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}])$$

by Bayes' rule:

$$p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]) = \frac{p(X(t_k^{\rm ms}), Z(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}])}{\int_{\mathcal{X}} p(X(t_k^{\rm ms}), Z(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms})}, \quad (32)$$

where $p(X(t_k^{\rm ms}), Z(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}])$ has the following form

$$p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}]) \times p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}]). \quad (33)$$

Since $Z(t_k^{\rm ms})$ is independent of $Z[t_{k-1}^{\rm ms}]$ and $W[t_{k-1}^{\rm ms}]$ when $X(t_k^{\rm ms})$ and $W(t_k^{\rm ms})$ are given, the first item in (33) can be written as $p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))$. For the second term in (33), it can be written as $p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])p(W[t_{k-1}^{\rm ms}]|W(t_k^{\rm ms}))$. Thus, (33) is rewritten as

$$p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) \cdot p(W[t_{k-1}^{\rm ms}]|W(t_k^{\rm ms})). \quad (34)$$

We put (34) back into (32), and (9) is derived [note that the common term $p(W[t_{k-1}^{\rm ms}]|W(t_k^{\rm ms}))$ in the numerator and denominator can be cancelled out].

ii) For (7), the proof is similar to that in (8). Using $X(t_k^{\rm ms})$ as the bridge, we have

$$\begin{aligned} &p(X(t)|Z[t], W[t]) \\ &\quad= p(X(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]) \\ &\quad= \int_{\mathcal{X}} p(X(t)|X(t_k^{\rm ms}))p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms}). \end{aligned}$$

APPENDIX B
PROOF OF THEOREM 1

We divide this proof into three parts: In the first two parts, we write $\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]]$ and $\mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})]$ into two suitable forms, respectively, so that they are comparable. In the third part, we complete this proof by comparing those two comparable forms.

i) By (5) and (7), $\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]]$ can be written as

$$\begin{aligned} &\int_{\mathcal{X}} g_l(x_l(t))\Big[\int_{\mathcal{X}} p(X(t)|X(t_k^{\rm ms})) \\ &\qquad\times p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms})\Big]{\rm d}X(t) \\ &= \int_{\mathcal{X}}\Big[\int_{\mathcal{X}} g_l(x_l(t))p(X(t)|X(t_k^{\rm ms})){\rm d}X(t)\Big] \\ &\qquad\times p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms}) \\ &= \int_{\mathcal{X}}\Big[\int_{\mathcal{X}_l} g_l(x_l(t))p(x_l(t)|X(t_k^{\rm ms})){\rm d}x_l(t)\Big] \\ &\qquad\times p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms}) \\ &\overset{(a)}{=} \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\big[g_l(x_l(t))|x_l(t_k^{\rm ms})\big] \\ &\qquad\times p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]){\rm d}X(t_k^{\rm ms}), \end{aligned} \quad (35)$$

where (a) follows from (19). We write term

$$p(X(t_k^{\rm ms})|Z[t_k^{\rm ms}], W[t_k^{\rm ms}])$$

in (35) as (9). Letting $F(X(t_k^{\rm ms})) = p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W[t_k^{\rm ms}])$ and $G_1(X(t_k^{\rm ms})) = p(X(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_k^{\rm ms}])$, we can re-express (35) as

$$\begin{aligned} &\mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]] \\ &\quad= \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\big[g_l(x_l(t))|x_l(t_k^{\rm ms})\big] \\ &\qquad\cdot \frac{F(X(t_k^{\rm ms}))G_1(X(t_k^{\rm ms}))}{\int_{\mathcal{X}} F(X(t_k^{\rm ms}))G_1(X(t_k^{\rm ms})){\rm d}X(t_k^{\rm ms})}{\rm d}X(t_k^{\rm ms}). \end{aligned} \quad (36)$$

ii) We rewrite the right-hand side of (10) as

$$\int_{\mathcal{X}_l} g_l(x_l(t))p(x_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})){\rm d}\bar{X}(t),$$

where term $p(x_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms}))$ is written as

$$\int_{\bar{\mathcal{X}}(t_k^{\rm ms})} p(x_l(t)|x_l(t_k^{\rm ms})) \cdot p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}]) d\bar{X}(t_k^{\rm ms}). \tag{37}$$

Since $\int_{\breve{\mathcal{X}}(t_k^{\rm ms})} p(\breve{X}(t_k^{\rm ms})|\breve{Z}[t_k^{\rm ms}], \breve{W}[t_k^{\rm ms}]) dX(t_k^{\rm ms}) \setminus \bar{X}(t_k^{\rm ms}) = 1$, where we have $\breve{\mathcal{X}}(t_k^{\rm ms}) = \mathcal{X} \setminus \bar{\mathcal{X}}(t_k^{\rm ms})$, $\breve{X}(t_k^{\rm ms}) = X(t_k^{\rm ms}) \setminus \bar{X}(t_k^{\rm ms})$, $\breve{Z}[t_k^{\rm ms}] = Z[t_k^{\rm ms}] \setminus \bar{Z}(t_k^{\rm ms})$, and $\breve{W}[t_k^{\rm ms}] = W[t_k^{\rm ms}] \setminus \bar{W}(t_k^{\rm ms})$, equation (37) can be further rewritten as

$$\int_{\mathcal{X}} p(x_l(t)|x_l(t_k^{\rm ms})) \cdot p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}]) \cdot p(\breve{X}(t_k^{\rm ms})|\breve{Z}[t_k^{\rm ms}], \breve{W}[t_k^{\rm ms}]) dX(t_k^{\rm ms}). \tag{38}$$

In (38), term $p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}])$ can be written as [similar to (9)]

$$\frac{p(\bar{Z}(t_k^{\rm ms})|\bar{X}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms}))p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])}{\text{Normalization}},$$

where

$$\text{Normalization} = \int_{\bar{\mathcal{X}}(t_k^{\rm ms})} p(\bar{Z}(t_k^{\rm ms})|\bar{X}(t_k^{\rm ms}), \bar{W}(t_k^{\rm ms})) \cdot p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) d\bar{X}(t_k^{\rm ms}).$$

Likewise, we can write term $p(\breve{X}(t_k^{\rm ms})|\breve{Z}[t_k^{\rm ms}], \breve{W}[t_k^{\rm ms}])$ in the integral of (37) as

$$\frac{p(\breve{Z}(t_k^{\rm ms})|\breve{X}(t_k^{\rm ms}), \breve{W}(t_k^{\rm ms}))p(\breve{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}])}{\text{Normalization}}, \tag{39}$$

where

$$\text{Normalization} = \int_{\breve{\mathcal{X}}(t_k^{\rm ms})} p(\breve{Z}(t_k^{\rm ms})|\breve{X}(t_k^{\rm ms}), \breve{W}(t_k^{\rm ms})) \cdot p(\breve{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}], W[t_{k-1}^{\rm ms}]) d\breve{X}(t_k^{\rm ms}).$$

In (39), $\breve{Z}(t_k^{\rm ms}) = Z(t_k^{\rm ms}) \setminus \bar{Z}(t_k^{\rm ms})$ and $\breve{W}(t_k^{\rm ms}) = W(t_k^{\rm ms}) \setminus \bar{W}(t_k^{\rm ms})$. Note that

$$\begin{aligned} &p(Z(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms})) \\ &= p(\bar{Z}(t_k^{\rm ms}), \breve{Z}(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms})) \\ &\overset{(b)}{=} p(\bar{Z}(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))p(\breve{Z}(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms})) \\ &\overset{(c)}{=} p(\bar{Z}(t_k^{\rm ms})|X(t_k^{\rm ms}), W(t_k^{\rm ms}))p(\breve{Z}(t_k^{\rm ms})|\breve{X}(t_k^{\rm ms}), W(t_k^{\rm ms})), \end{aligned} \tag{40}$$

where (b) holds with the independence of the measurement noises in (3), and (c) follows from condition (13) in Definition 2. The product of the first items in the numerators of (39) and (40) is $p(Z(t_k^{\rm ms})|X(t_k^{\rm ms})) = F(X(t_k^{\rm ms}))$. Letting

$$G_2(X(t_k^{\rm ms})) = p(\bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}])p(X(t_k^{\rm ms}) \setminus \bar{X}(t_k^{\rm ms})|Z[t_{k-1}^{\rm ms}]),$$

i.e. the product of the second items in the numerators of (39) and (40), we can write the estimation $\mathbb{E}_{y_l(t)}^{\bar{X}(t)}\left[y_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})\right]$ as

$$\begin{aligned} &\int_{\mathcal{X}_l} g_l(x_l(t)) \left[\int_{\mathcal{X}} p(x_l(t)|x_l(t_k^{\rm ms}))FG_2(X(t_k^{\rm ms}))dX(t_k^{\rm ms})\right] dx_l(t) \\ &= \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\rm ms})\right]FG_2(X(t_k^{\rm ms}))dX(t_k^{\rm ms}), \end{aligned} \tag{41}$$

where

$$FG_2(X(t_k^{\rm ms})) := \frac{F(X(t_k^{\rm ms}))G_2(X(t_k^{\rm ms}))}{\int_{\mathcal{X}} F(X(t_k^{\rm ms}))G_2(X(t_k^{\rm ms}))dX(t_k^{\rm ms})}.$$

We can see that (36) and (41) have a similar structure. Actually, the only difference comes from $G_1(X(t_k^{\rm ms}))$ and $G_2(X(t_k^{\rm ms}))$.

iii) Subtracting (41) from (36), we have

$$\begin{aligned} &\mathbb{E}_{y_l(t)}\left[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]\right] \\ &\quad - \mathbb{E}_{y_l(t)}^{\bar{X}(t)}\left[y_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})\right] \\ &= \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\rm ms})\right]F(X(t_k^{\rm ms})) \\ &\quad \cdot \frac{H_2\, G_1(X(t_k^{\rm ms})) - H_1\, G_2(X(t_k^{\rm ms}))}{H_1\, H_2} dX(t_k^{\rm ms}), \end{aligned} \tag{42}$$

where

$$\begin{aligned} H_1 &= \int_{\mathcal{X}} F(X(t_k^{\rm ms}))G_1(X(t_k^{\rm ms}))dX(t_k^{\rm ms}), \\ H_2 &= \int_{\mathcal{X}} F(X(t_k^{\rm ms}))G_2(X(t_k^{\rm ms}))dX(t_k^{\rm ms}). \end{aligned} \tag{43}$$

Note that $H_1, H_2 > 0$. We split (42) into two parts, i.e.

$$\begin{aligned} &\mathbb{E}_{y_l(t)}\left[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]\right] \\ &\quad - \mathbb{E}_{y_l(t)}^{\bar{X}(t)}\left[y_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})\right] \\ &= A + B, \end{aligned}$$

where

$$A = \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\rm ms})\right] \cdot F(X(t_k^{\rm ms}))\frac{G_1(X(t_k^{\rm ms})) - G_2(X(t_k^{\rm ms}))}{H_1} dX(t_k^{\rm ms}), \tag{44}$$

$$B = \frac{H_2 - H_1}{H_1\, H_2} \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}\left[g_l(x_l(t))|x_l(t_k^{\rm ms})\right] \cdot F(X(t_k^{\rm ms}))G_1(X(t_k^{\rm ms}))dX(t_k^{\rm ms}). \tag{45}$$

In the rest of this proof, we find the upper bounds for $\|A\|$ and $\|B\|$, respectively, so that the following can be upper bounded

$$\begin{aligned} &\Big\| \mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\rm ms}], W[t_k^{\rm ms}]] \\ &\quad - \mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\rm ms}], \bar{Z}(t_k^{\rm ms}), W[t_{k-1}^{\rm ms}], \bar{W}(t_k^{\rm ms})] \Big\| \\ &\leq \|A\| + \|B\|. \end{aligned} \tag{46}$$

For A, we have (47), as shown at the top of the next page. where (d) follows from the "triangle inequality" and

$$
\begin{aligned}
\|A\| &= \left\| \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] F(X(t_k^{\text{ms}})) \frac{G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))}{H_1} \mathrm{d}X(t_k^{\text{ms}}) \right\| \\
&\overset{(d)}{\leq} \int_{\mathcal{X}} \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})]) \right\| F(X(t_k^{\text{ms}})) \left| \frac{G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))}{H_1} \right| \mathrm{d}X(t_k^{\text{ms}}) \\
&= \left\| \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] \right\| F(X(t_k^{\text{ms}})) \frac{G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))}{H_1} \right\|_1 \\
&\overset{(e)}{=} \left\| A_1(X(t_k^{\text{ms}})) \frac{G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}}))}{H_1} \right\|_1,
\end{aligned} \tag{47}
$$

$$
\begin{aligned}
\|B\| &= \frac{|H_2 - H_1|}{H_2} \left\| \int_{\mathcal{X}} \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \right\| \\
&\leq \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] \right\| p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\overset{(f)}{=} \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left\| \int_{\mathcal{X}_l} g(x_l(t)) p(x_l(t)|x_l(t_k^{\text{ms}})) \mathrm{d}x_l(t) \right\| p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\leq \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}} \left[\int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(x_l(t)|x_l(t_k^{\text{ms}})) \mathrm{d}x_l(t) \right] p(X(t_k^{\text{ms}})|Z[t_k^{\text{ms}}]) \mathrm{d}X(t_k^{\text{ms}}) \\
&\overset{(g)}{=} \frac{|H_2 - H_1|}{H_2} \int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(X(t)|Z[t_k^{\text{ms}}]) \mathrm{d}x_l(t) \\
&\overset{(h)}{=} V \frac{|H_2 - H_1|}{H_2},
\end{aligned} \tag{48}
$$

$F(X(t_k^{\text{ms}})) \geq 0$. For (e), it holds with $A_1(X(t_k^{\text{ms}})) = \left\| \mathbb{E}_{x_l(t)}[g_l(x_l(t))|x_l(t_k^{\text{ms}})] \right\| F(X(t_k^{\text{ms}}))$.

With Hölder's inequality, (47) can be zoomed as

$$
\|A\| \leq \left\| A_1(X(t_k^{\text{ms}})) \right\|_1 \left\| G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}})) \right\|_\infty . \tag{49}
$$

From (17), we know that $\left\| A_1(X(t_k^{\text{ms}})) \right\|_1$ is bounded. Since the cut gap defined in (16) is not greater than δ, inequality (49) can be rewritten as

$$
|A| \leq \left\| A_1(X(t_k^{\text{ms}})) \right\|_1 \delta.
$$

For B, from (45), we have (48), as shown at the top of this page, where (f) follows from (19), and (g) is based on the interchange of the order of integration.

According to (6), we set

$$
\int_{\mathcal{X}_l} \|g(x_l(t))\| \, p(X(t)|Z[t_k^{\text{ms}}]) \mathrm{d}x_l(t) = \int_{\mathcal{X}} \|g_l(x_l(t))\| p(X(t)|Z[t], W[t]) \mathrm{d}X(t) =: V,
$$

which leads to (h). According to (43), we rewrite term $|H_1 - H_2|$ in (48) as

$$
\begin{aligned}
|H_1 - H_2| &= \left| \int_{\mathcal{X}} F(X(t_k^{\text{ms}})) \left[G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}})) \right] \mathrm{d}X(t_k^{\text{ms}}) \right| \\
&\overset{(i)}{\leq} \left\| F(X(t_k^{\text{ms}})) \right\|_1 \left\| G_1(X(t_k^{\text{ms}})) - G_2(X(t_k^{\text{ms}})) \right\|_\infty \\
&\leq \left\| F(X(t_k^{\text{ms}})) \right\|_1 \delta,
\end{aligned}
$$

where (i) follows from Hölder's inequality. Thus, (48) can be rewritten as

$$
\|B\| \leq V \frac{\left\| F(X(t_k^{\text{ms}})) \right\|_1}{H_2} \delta. \tag{50}
$$

Note that $\left\| F(X(t_k^{\text{ms}})) \right\|_1$ is bounded by (18). Combing (46), (47), and (50), we have

$$
\begin{aligned}
&\Big\| \mathbb{E}_{y_l(t)}[y_l(t)|Z[t_k^{\text{ms}}], W[t_k^{\text{ms}}]] \\
&\quad - \mathbb{E}_{y_l(t)}^{\bar{X}(t)}[y_l(t)|Z[t_{k-1}^{\text{ms}}], \bar{Z}(t_k^{\text{ms}}), W[t_{k-1}^{\text{ms}}], \bar{W}(t_k^{\text{ms}})] \Big\| \\
&\leq \left[\left\| A_1(X(t_k^{\text{ms}})) \right\|_1 + V \frac{\left\| F(X(t_k^{\text{ms}})) \right\|_1}{H_2} \right] \delta.
\end{aligned}
$$

Therefore, $\forall \varepsilon > 0$, there exists a

$$
\delta \leq \frac{H_2}{H_2 \left\| A_1(X(t_k^{\text{ms}})) \right\|_1 + V \left\| F(X(t_k^{\text{ms}})) \right\|_1} \varepsilon, \tag{51}
$$

such that if the cut gap is bounded by δ in (51), then the estimation gap defined in (11) is bounded by ε.

APPENDIX C
COMMUNICATION SETTINGS IN SECTION VI-A3

The transmit power of non-anchor node $l \in \mathcal{L}$ is $P_l^{\text{tr}}(t)$ with maximum value $P_{\max}^{\text{tr}} = 0.2\text{W}$, and the power of noise (from the receiver side) is $P_{\text{noise}} = -90\text{dBm}$. The relationship between the received and transmit powers at time $t \in [0,\ 50]$ is

$$
P_{j,l}^{\text{rc}}(t) = \frac{h_{l,j}(t) P_l^{\text{tr}}(t)}{1 + D_{l,j}^{\alpha}(t)},
$$

where $h_{l,j}(t)$ is the fading gain with Nakagami-m model (see [20]) and $m = 2$. Notation $D_{l,j}(t)$ is the distance between nodes l and j, and α is the path loss exponent. In this problem, we assume $\alpha = 3$.

Note that there are 20 non-anchor nodes sharing the same bandwidth, and each broadcast can generate interferences. To mitigate the interference, we use the CSMA technique. The threshold for detecting an idle channel is determined by the following equation

$$\theta = \frac{P_{\max}^{\text{tr}}}{1 + D_*^\alpha},$$

where parameter D_* is the silence distance that if a node is outside the range covered by this distance, then the transmission interference can be neglected. We set $D_* = 2$m in this problem, which means the node only regards the communication signals within 2 meters as interferences. If the received power is lower than θ, then a node assumes this channel is idle for broadcasting, and it uses the maximum transmit power to transmit (on-off power control), otherwise the node would wait for a random time follows the exponential distribution with mean $\mu = 0.05$s.

For message receiving, the SINR for node l receiving the message from node j is

$$\text{SINR}_{l,j}(t) = \frac{P_{l,j}^{\text{rc}}(t)}{\sum_{i \neq j} P_{l,i}^{\text{rc}}(t) + P_{\text{noise}}}.$$

The channel is complex Gaussian, and the Shannon's capacity is

$$C_{l,j}(t) = B \log_2 \left(1 + \text{SINR}_{l,j}(t)\right),$$

where the bandwidth is $B = 40$MHz. The broadcasting rate for node $j \in \mathcal{L}$ is $R_j^{\text{bc}} = 100\text{KB/s} = 800\text{Kb/s}$. If the average capacity for a broadcast message is lower than the broadcasting rate, then a transmission outage occurs. Specifically, an outage happens when

$$\frac{1}{t_{j,k}^{\text{et}} - t_{j,k}^{\text{st}}} \int_{t_{j,k}^{\text{st}}}^{t_{j,k}^{\text{et}}} C_{l,j}(t)\text{d}t < R_j^{\text{bc}},$$

where $t_{j,k}^{\text{st}}$ and $t_{j,k}^{\text{et}}$ the start and end times, respectively, of broadcast message $m_{j,k}^{\text{bc}}$ (see Fig. 2). The broadcasting time length $t_{j,k}^{\text{et}} - t_{j,k}^{\text{st}}$ depends on the packet size of $m_{j,k}^{\text{bc}}$, i.e.

$$t_{j,k}^{\text{et}} - t_{j,k}^{\text{st}} = \frac{\text{size}(m_{j,k}^{\text{bc}})}{R_j^{\text{bc}}}.$$

The packet size is determined by the content in the broadcasting message, where the real numbers (e.g., for describing the priors, posteriors, and measurements) are stored in double precision numbers with 64bit size.

REFERENCES

[1] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[2] M. Z. Win *et al.*, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[3] W. Yuan, N. Wu, B. Etzlinger, H. Wang, and J. Kuang, "Cooperative joint localization and clock synchronization based on Gaussian message passing in asynchronous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7258–7273, Sep. 2016.

[4] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, and H. Wymeersch, "Cooperative simultaneous localization and synchronization in mobile agent networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3587–3602, Jul. 2017.

[5] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[6] S. Särkkä, *Bayesian Filtering and Smoothing*, vol. 3. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[7] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.

[8] V. Savic and S. Zazo, "Reducing communication overhead for cooperative localization using nonparametric belief propagation," *IEEE Wireless Commun. Lett.*, vol. 1, no. 4, pp. 308–311, Aug. 2012.

[9] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 145–149, Feb. 2014.

[10] S. Van de Velde, G. T. F. de Abreu, and H. Steendam, "Improved censoring and NLOS avoidance for wireless localization in dense networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2302–2312, Nov. 2015.

[11] C. Pedersen, T. Pedersen, and B. H. Fleury, "A variational message passing algorithm for sensor self-localization in wireless networks," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 2158–2162.

[12] B. Çakmak, D. N. Urup, F. Meyer, T. Pedersen, B. H. Fleury, and F. Hlawatsch, "Cooperative localization for mobile networks: A distributed belief propagation–mean field message passing algorithm," *IEEE Signal Process. Lett.*, vol. 23, no. 6, pp. 828–832, Jun. 2016.

[13] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, D, J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[14] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY, USA: Academic, 1970.

[15] S. Särkkä, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1631–1641, Sep. 2007.

[16] I. Arasaratnam, S. Haykin, and T. R. Hurd, "Cubature Kalman filtering for continuous-discrete systems: Theory and simulations," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 4977–4993, Oct. 2010.

[17] S. Särkkä and T. Sottinen, "Application of Girsanov theorem to particle filtering of discretely observed continuous-time non-linear systems," *Bayesian Anal.*, vol. 3, no. 3, pp. 555–584, 2008.

[18] B. Øksendal, *Stochastic Differential Equations: An Introduction With Applications*, 5th ed. Berlin, Germany: Springer-Verlag, 2000.

[19] F. Bai and A. Helmy, *A Survey of Mobility Models* (Wireless Ad-Hoc Networks), vol. 206. Los Angeles, CA, USA: Univ. Southern California, Jun. 2004, ch. 1, pp. 1–30.

[20] Y. Cong, X. Zhou, and R. A. Kennedy, "Interference prediction in mobile ad hoc networks with a general mobility model," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4277–4290, Aug. 2015.

[21] D. J. Higham, "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM Rev.*, vol. 43, no. 3, pp. 525–546, Mar. 2001.

[22] L. Murray and A. Storkey, "Particle smoothing in continuous time: A fast approach via density estimation," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1017–1026, Mar. 2011.

[23] Y. Xia, Z. Deng, L. Li, and X. Geng, "A new continuous-discrete particle filter for continuous-discrete nonlinear systems," *Inf. Sci.*, vol. 242, pp. 64–75, Sep. 2013.

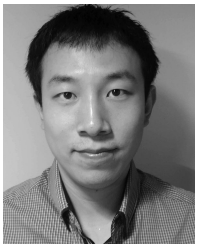
[24] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual*. Reading, MA, USA: Addison-Wesley, 2001.

[25] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2409–2419, Mar. 2018.

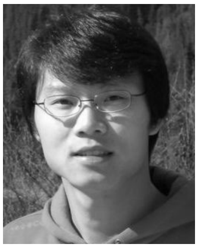
[26] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li, "One more tag enables fine-grained RFID localization and tracking," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 161–174, Feb. 2018.

[27] K. M. Nagpal and P. P. Khargonekar, "Filtering and smoothing in an H^∞ setting," *IEEE Trans. Autom. Control*, vol. 36, no. 2, pp. 152–166, Feb. 1991.

[28] J. Liu, C. Wu, Z. Wang, and L. Wu, "Reliable filter design for sensor networks using type-2 fuzzy framework," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1742–1752, Aug. 2017.

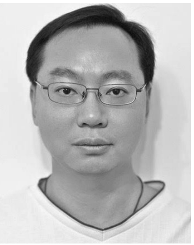


YIRUI CONG (S'14–M'18) received the B.E. degree (Hons.) in automation from Northeastern University, Shenyang, China, in 2011, the M.Sc. degree (graduated in advance) in control science and engineering from the National University of Defense Technology, Changsha, China, in 2013, and the Ph.D. degree from Australian National University, Canberra, ACT, Australia, in 2018. He is currently a Lecturer with the National University of Defense Technology. His research interests are in the fields of communication theory and control theory.



XIANGYUN ZHOU (M'11–SM'17) received the Ph.D. degree from Australian National University (ANU), in 2010. He is currently an Associate Professor with ANU. His research interests are in the general fields of communication theory and networks. He currently serves on the editorial board for the IEEE Transactions on Wireless Communications and the IEEE Wireless Communications Letters. He was the Chair of the ACT Chapter of the IEEE Communications Society and the Signal Processing Society from 2013 to 2014.

BINGPENG ZHOU (S'16–M'17) received the B.Eng. degree from the Zhongyuan University of Technology, China, in 2010, and the Ph.D. degree from Southwest Jiaotong University, China, in 2016, respectively. He currently is a Post-Doctoral Fellow with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST). Prior to this, he has been a Visiting Scholar with HKUST for six months. He was a Visiting Ph.D. Student at the 5G Innovation Centre, University of Surrey, Guildford, U.K., for three months in 2015. His current research interests include wireless localization and tracking, Bayesian signal processing, and vehicular ad-hoc networks.



VINCENT K. N. LAU (SM'04–F'12) received the B.Eng. degree (Hons.) from the University of Hong Kong in 1992 and the Ph.D. degree from Cambridge University in 1997. He joined Bell Labs from 1997 to 2004 and the Department of ECE, Hong Kong University of Science and Technology (HKUST) in 2004. He is currently a Chair Professor and the Founding Director of the Huawei-HKUST Joint Innovation Laboratory, HKUST. He is also elected as the HKIE Fellow, Croucher Senior Research Fellow, and Changjiang Chair Professor. He has authored or co-authored over 300 IEEE journal and conference papers and has contributed to 50 U.S. patents on various wireless systems. His current research interests include stochastic optimization and analysis for wireless systems, massive MIMO, compressed sensing, networked control systems, and PHY caching for wireless networks.