Distributed Finite-Time Cooperative Localization for Three-Dimensional Sensor Networks

Distributed Finite-Time Cooperative Localization for Three-Dimensional Sensor Networks

Jinze Wu, Lorenzo Zino, *Senior Member, IEEE,* Zhiyun Lin, *Fellow, IEEE*, and Alessandro Rizzo, *Senior Member, IEEE*

***Abstract*—This paper addresses the distributed localization problem for a network of sensors placed in a three-dimensional space, in which sensors are able to perform range measurements, i.e., measure the relative distance between them, and exchange information on a network structure. While most existing studies primarily develop localization algorithms under the assumption that the entire sensor network is localizable, the problem of determining whether a sensor is localizable has received limited attention. However, neglecting this preliminary step can significantly hamper the accuracy of localization algorithms due to error propagation from unlocalizable sensors in iterative localization procedures. To address this research gap, we focus on two key challenges: i) deriving rigorous theoretical results and developing algorithms to verify sensor localizability, and ii) designing an efficient distributed localization algorithm that utilizes these localizability results. Specifically, we start by deriving a necessary and sufficient condition for sensor localizability using barycentric coordinates. Then, building on this theoretical result, we design a distributed localizability verification algorithm, in which we propose and employ a novel distributed finite-time algorithm for sum consensus. Finally, we develop a distributed localization algorithm based on conjugate gradient method and derive theoretical guarantees on its performance, ensuring finite-time convergence. The efficiency of our algorithm compared to the existing ones from the literature and its capability to handle scenarios with moderate levels of noise in the measurements are further demonstrated through numerical simulations.**

***Index Terms*—Distributed localization, sensor network, node localizability, conjugate gradient.**

I. Introduction

LOCALIZATION has become a critical issue as the position information of a set of agents plays an increasingly important role in multiple applications, such as target searching, environmental monitoring, navigation, and formation control [1]–[7]. The most common method to obtain position information is to equip all agents with GPS, a satellite-based global positioning system that provides the coordinates of receivers [8]. However, GPS has some drawbacks in terms of coverage and power consumption [9]. More specifically, it may not work properly in environments with obstructions between the GPS satellites and the receivers, e.g., interior buildings and areas with dense vegetation or mountains. Furthermore, a scenario in which all agents are equipped with GPS consumes more energy than one with only a few of them equipped with GPS. Thus, cooperative localization based on terrestrial techniques such as cellular, WiFi, and ultra-wideband has come to the fore [10]–[14].

The objective of cooperative localization is to determine the Euclidean coordinates of all agents given the Euclidean coordinates of a (small) subset of reference agents, and assuming that the rest of the agents are able to make local measurements (e.g., the relative distance, bearing, or angle between them) and exchange information over a network. For this reason, in the rest of this paper we will refer to the set of agents as a *sensor network*. Given the importance of this problem, several approaches have been proposed in the literature.

Since the essence of the cooperative localization problem is nonlinear (being the relationship between Euclidean coordinates and local measurements generally nonlinear), most authors in the literature focused on nonlinear algorithms, formulating the cooperative localization problem as a constrained nonlinear optimization problem [15]–[18]. However, these approaches have several crucial limitations. First, solvers for such nonlinear optimization problems typically lack theoretical guarantees to ensure global convergence [16]. Second, as the scale of the sensor network grows, solving nonlinear equations results in an increased number of saddle points due to iterations, limiting the possibility to implement them in large-scale real-world applications [19]. Third, most of the nonlinear localization algorithms proposed in the literature need to be solved in a centralized fashion, yielding a more complex and less robust architecture.

In order to address the limitations of nonlinear algorithms, distributed linear cooperative localization algorithms have attracted a lot of research interest in recent years. In particular, different algorithms have been proposed to tackle this problem, depending on the type of local measurements that the sensors are able to perform, including range-based [20]–[24], bearing-based [25]–[28], relative position-based [29], [30], angle-based [31]–[33] and mixed measurements-based [34], [35] methods.

A common technique to represent space in these efforts is the use of barycentric coordinates [36] –a coordinate system in which the location of a point is specified with reference

J. Wu, L. Zino, and A. Rizzo are with the Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy. Emails: {jinze.wu,lorenzo.zino,alessandro.rizzo}@polito.it.
J. Wu and Z. Lin are with Guangdong Provincial Key Laboratory of Fully Actuated System Control Theory and Technology, School of Automation and Intelligent Manufacturing, Southern University of Science and Technology, Shenzhen, China. Email: linzy@sustech.edu.cn. Z. Lin is also with Peng Cheng Laboratory, Shenzhen, China. This work was partially supported by National Natural Science Foundation of China, under Grant No. 62173118; Guangdong Science and Technology Program under Grant No. 2024B1212010002; Shenzhen Science and Technology Program under Grant No. KQTD20221101093557010; and FAIR — Future Artificial Intelligence Research, and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) — MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). *Corresponding authors: Z. Lin and A. Rizzo.*

to other points. Barycentric coordinate systems have emerged as one of the most effective tools to obtain the prerequisites for addressing the cooperative localization problem in a distributed and linear manner. The authors in [20] proposed a distributed localization algorithm based on barycentric coordinates that uses range-based measurements (i.e., in which it is assumed that sensors can measure the relative distance between them). This algorithm is implemented by expressing the positions of the sensors as a pseudo linear system with all nonlinearities hidden in range measurements and relies on the assumption that each sensor requires to lie inside the convex hull of its neighbors. To drop this assumption, [22] introduced a generalized barycentric coordinates approach, deriving barycentric coordinates by linking distance measurements with coordinate signs. However, unlike [20], this formulation may result in unstable eigenvalues, complicating distributed and iterative computations. To overcome this issue, they developed a novel localization algorithm in a two-dimensional space. Building on this approach, [23] presented a more robust localization algorithm in a two-dimensional space, adopting the idea of the congruent framework. Such approach was successfully extended to a three-dimensional space in [24]. Besides range-based measurements, barycentric coordinates were also employed to solve localization problems in other scenarios, with bearing [26], [28], relative position [30], angle [31] and mixed [34] measurements. Moreover, the angle-based localization algorithms proposed in [32], [33] can be converted into barycentric coordinates for solution as well.

In addition to designing localization algorithms, a problem of paramount importance is to determine which sensors are localizable, given the specific characteristics of the sensor network setup (i.e., topology of the sensing and communication channel, and type of measurement that sensor can perform) [37], [38]. Most of the aforementioned works have focused on proposing localization algorithms under the assumption that all sensors can be localized in the sensor network. In other words, these works solely address network localizability without delving into the localizability of individual sensors. However, neglecting whether a sensor is localizable or not can lead to considerable errors in the localization process of all sensors. Not only do unlocalizable sensors fail to estimate their correct positions, but the locations of the localizable sensors will also be estimated inaccurately or to completely incorrect positions due to the iterative propagation of wrong positions from unlocalizable sensors. Therefore, it is crucial to identify all localizable and unlocalizable sensors and filter out the unlocalizable sensors before implementing a distributed localization algorithm. Accurate identification of localizable sensors is a key step to ensure the reliability and precision of the localization process in a sensor network.

In this paper, we address the cooperative localization problem, without making any a priori assumption on localizability. Specifically, we deal with the sensor localizability and the range-based localization problem for sensor networks in a three-dimensional space. The key contribution of this work lies in the integration of the localizability verification process and the localization process: without the localizability verification process, the localization results may be misled by unlocalizable sensors, which hampers the correct application of any localization method. In this context, the main contribution of this paper is threefold. First, we prove a necessary and sufficient condition for sensor localizability for the range-based localization problem in a three-dimensional space using barycentric coordinates, and then propose a distributed verification algorithm to distinguish between localizable and unlocalizable sensors in the sensor network. It is worth mentioning that these tools can be readily generalized to various barycentric-coordinate localization problems, establishing them as a universal sensor localizability verification approach, which may be implemented to lift the restrictions of other localization algorithms in the presence of unlocalizable sensors. Second, we develop a distributed conjugate gradient localization algorithm to efficiently solve the localization problem for the sensor network in finite time. Its finite-time convergence is proved theoretically, and then validated in numerical simulations. Third, in the development of our algorithms, we propose a novel distributed algorithm, which is able to achieve sum consensus among all sensors in the network in finite time. Such algorithm, which is used in our localization process, is a distributed consensus algorithm that considers the tradeoff between memory size and iteration steps, and can be directly employed in other application fields.

The rest of the paper is organized as follows. Section II presents notation and preliminaries. In Section III, we formulate the research problem. In Section IV, we propose a distributed algorithm to address the localizability verification problem. Section V is devoted to the distributed localization algorithm and the proof of its finite-time convergence. Numerical simulations are presented in Section VI. Section VII concludes the paper and outlines future research directions.

II. Notation and Preliminaries

A. Notation

We use uppercase letters for matrices, bold lowercase letters for vectors, and lowercase letters for scalars. With $\mathbf{x} \in \mathbb{R}^n$ we refer to a column vector, and we use the transpose operator ($\mathbf{x}^\top$) to denote row vectors. Let $\mathbb{R}^n$ be the n-dimensional real coordinate space. I_n denotes the identity matrix of order $n \times n$, while $\mathbf{1}_n$ denotes the n-dimensional vector with all entries equal to 1 and $\mathbf{e_i}$ denotes a vector whose ith entry is 1 and all other entries are 0. The notation $|\cdot|$ denotes the absolute value of a constant, the cardinality of a set, or the determinant of a matrix, depending on the context. The symbol $\|\cdot\|$ denotes the 2-norm and $\otimes$ denotes the Kronecker product. Moreover, the symbol $\lceil x \rceil$ refers to the smallest integer that is larger than or equal to a real number x. Finally, let $\mathrm{diag}(\cdot)$ denote the diagonal matrix. Given a matrix $A \in \mathbb{R}^{n\times m}$, $\mathrm{rank}(A)$ and $\ker(A)$ are its rank and kernel, respectively.

B. Graph Theory

An *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a nonempty set $\mathcal{V}$, whose elements are called *nodes* or *vertices*, and a set of unordered pairs of nodes $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, whose elements are called *edges*. For a node i, we define its *neighbor set* as $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Clearly, if $i \in \mathcal{N}_j$ then $j \in \mathcal{N}_i$. A

path of length ℓ from $i \in \mathcal{V}$ to $j \in \mathcal{V}$ is a sequence of ℓ edges of the form $(i, v_1), (v_1, v_2), \ldots, (v_{\ell-1}, j)$. A graph is said to be *connected* if there is a path between every pair of vertices. The *diameter* of a connected graph, denoted by δ, is the maximum among all the lengths of the shortest path between any pair of vertices of the graph. A subset of d vertices such that each and every pair of them are connected through an edge is said to be a *clique* of order d. Here, we shall refer to cliques of order 4 as *tetrahedrons* and use the notation $\triangle ijkl$ for a tetrahedron with vertices $i, j, k, l \in \mathcal{V}$. Note that a clique of order 5 is formed by 5 tetrahedrons.

Given a set of n nodes positioned in the Euclidean space $\mathbb{R}^3$, we define their *configuration* as the vector $\mathbf{p} = [\mathbf{p}_1^\top, \cdots, \mathbf{p}_n^\top]^\top \in \mathbb{R}^{3n}$, where the entry $\mathbf{p_i} \in \mathbb{R}^3$ is the position of node i in the three-dimensional Euclidean space. A configuration $\mathbf{p} = [\mathbf{p}_1^\top, \cdots, \mathbf{p}_n^\top]^\top \in \mathbb{R}^{3n}$ is said to be *generic* if the coordinates $\mathbf{p_1}, \cdots, \mathbf{p_n}$ do not satisfy any nonzero polynomial equation with integer coefficients or equivalently algebraic coefficients [39]. In other words, a generic configuration has no degeneracy, that is, no three points staying on the same line, no three lines go through the same point, no four points staying on the same plane, etc.

A *framework* $\mathcal{F} = (\mathcal{G}, \mathbf{p})$ in the Euclidean space $\mathbb{R}^3$ is formed by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a configuration $\mathbf{p} \in \mathbb{R}^{3n}$. Two frameworks with the same graph $(\mathcal{G}, \mathbf{p})$ and $(\mathcal{G}, \mathbf{q})$ are said to be *congruent* (denoted by $(\mathcal{G}, \mathbf{p}) \equiv (\mathcal{G}, \mathbf{q})$) if and only if (iff) $\|\mathbf{p_i} - \mathbf{p_j}\| = \|\mathbf{q_i} - \mathbf{q_j}\|$ holds for all pairs $i, j \in \mathcal{V}$. In other words, all nodes of two congruent frameworks maintain the same distances between pairs. It is easy to check that the property of being congruent is an equivalence relation. It is important to notice that such an equivalence relation preserves volumes in the space.

C. Barycentric Coordinates

The *barycentric coordinates* were introduced by A. F. Möbius in 1827 as mass points to define a coordinate-free geometry [36]. In fact, barycentric coordinates characterize the relative position of a node with respect to other nodes. Specifically, given five nodes i, j, k, l, and h with their Euclidean coordinates $\mathbf{p_i}, \mathbf{p_j}, \mathbf{p_k}, \mathbf{p_l}, \mathbf{p_h} \in \mathbb{R}^3$, the barycentric coordinates of node i with respect to nodes j, k, l, and h are equal to the quadruple $\{a_{ij}, a_{ik}, a_{il}, a_{ih}\}$, solution of the following system of equations:

$$\begin{cases} \mathbf{p_i} = a_{ij}\mathbf{p_j} + a_{ik}\mathbf{p_k} + a_{il}\mathbf{p_l} + a_{ih}\mathbf{p_h}, \\ a_{ij} + a_{ik} + a_{il} + a_{ih} = 1. \end{cases} \tag{1}$$

In other words, using the barycentric coordinates, we represent the position of a point in a three-dimensional space as a combination of the positions of four other points in the space, and we use the weights of such combination to identify the point. Fig. 1 illustrates a graphic representation of the barycentric coordinates.

The barycentric coordinates of node i can be calculated using the signed volumes between corresponding tetrahedrons [36]. Specifically, it holds

$$a_{ij} = \frac{V_{iklh}}{V_{jklh}}, a_{ik} = \frac{V_{jilh}}{V_{jklh}}, a_{il} = \frac{V_{jkih}}{V_{jklh}}, a_{ih} = \frac{V_{jkli}}{V_{jklh}}, \tag{2}$$

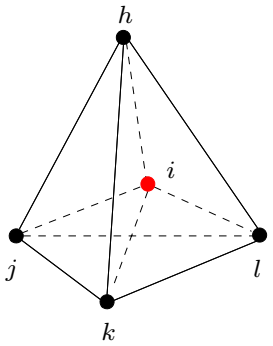

Fig. 1. Illustration of the barycentric coordinates of node i with respect to nodes j, k, l, and h in a three-dimensional space.

where $V_{iklh}, V_{jilh}, V_{jkih}, V_{jkli}$ and V_{jklh} are the signed volumes of the tetrahedrons $\triangle iklh$, $\triangle jilh$, $\triangle jkih$, $\triangle jkli$, and $\triangle jklh$, respectively. The signed volume V_{iklh} is equal in modulus to the volume of the tetrahedron $\triangle iklh$, with positive sign if nodes i, k, l, and h conform to the right hand corkscrew rule, and negative otherwise. Given the coordinates of its four vertices, the signed volume V_{iklh} is equal to the following determinant of a 4×4 matrix:

$$V_{iklh} = \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 & 1 \\ \mathbf{p_i} & \mathbf{p_k} & \mathbf{p_l} & \mathbf{p_h} \end{vmatrix}. \tag{3}$$

Note that, due to the volume conservation law under roto-translation, the barycentric coordinates of a node are invariant with respect to congruent frameworks.

III. Problem Setup and Statement

In this section, we describe the setup of the localization problem and formulate the problem statement.

A. Problem Setup

Consider n sensors positioned in a three-dimensional Euclidean space. Sensors are allowed to interact with each others —measuring their relative distance (range measurement)— and communicate, subject to some constraints. Specifically, we define a *sensor network* that is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by associating a node of the graph with each sensor, and edges to describe the sensing and communication topology of the sensor network. That is, we assume that $(i, j) \in \mathcal{E}$ iff sensors i and j can measure their relative distance and communicate, i.e., we assume that the sensing and communication graphs of the sensor network are the same and that the network is connected.

The sensor network consists of a small number of sensors with known position and a large number of sensors with unknown position. The sensors whose position are known (e.g., obtained through GPS or being able to self-calibrate) are called *anchor nodes*. In the theoretical derivations of this paper, we will assume that anchor nodes know their exact positions. The rest of the sensors are called *free nodes*. The objective of cooperative localization is to estimate the Euclidean coordinates in the global coordinate system of the free nodes in a distributed fashion by exchanging information on their range

measurements, given the absolute Euclidean coordinates of the anchor nodes. To this aim, nodes can make measurements of their relative distance (range measurements) and they can communicate with their neighbors on the network.

We observe that a necessary condition for all free nodes to be localized in a three-dimensional space using relative measurements is that the sensor network has at least four anchor nodes [40]. Hence, without any loss in generality, we assume there are four anchor nodes and $n-4$ free nodes. We denote these sets as $\mathcal{V} = \mathcal{V}_f \cup \mathcal{V}_a$, with $\mathcal{V}_f = \{1, \cdots, n-4\}$ and $\mathcal{V}_a = \{n-3, \cdots, n\}$ denoting the free node set and the anchor node set, respectively. If the sensor network has the edge $(i,j) \in \mathcal{E}$, then node i can measure the range measurement $d_{ij} = \|\mathbf{p_i} - \mathbf{p_j}\|$ and exchange such information with node j. Here, we will assume that such measurements are exact, i.e., they are not subject to noises. The robustness of the proposed method with respect to noise in the measurements will be then demonstrated via numerical simulations. Let $\mathbf{p_f} = [\mathbf{p_1}^\top, \cdots, \mathbf{p_{n-4}}^\top]^\top \in \mathbb{R}^{3(n-4)}$ and $\mathbf{p_a} = [\mathbf{p_{n-3}}^\top, \cdots, \mathbf{p_n}^\top]^\top \in \mathbb{R}^{12}$ denote the Euclidean coordinate vectors of the free nodes and of the anchor nodes, respectively. It is clear that $\mathbf{p_a}$ is known, but the Euclidean coordinates of free node set $\mathbf{p_f}$ needs to be estimated.

All free nodes in the network can be characterized depending on their possibility to be localized in a distributed fashion using barycentric coordinates. Hence, we define two different classes: *localizable* nodes and *unlocalizable* nodes. According to [40], a node requires at least four neighbors for localizability in a three-dimensional space. Moreover, in order to use barycentric coordinates, a node $i \in \mathcal{V}$ should position itself with respect to other 4 nodes. Intuitively, we need these 5 nodes (i.e., node i and the four neighbors) to be able to communicate in order to compute the volumes of the 5 tetrahedrons they form and, ultimately, the barycentric coordinates of i. Hence, node i should belong to a clique of order 5 to be localized. As a consequence, all nodes that do not belong to any clique of order 5 belong necessarily to the class of unlocalizable nodes, which we term *scarcely connected* nodes.

Remark 1: Node i has information on its neighbor set $\mathcal{N}_i$. Moreover, i can exchange such information with its neighbors. Hence, for any $j \in \mathcal{N}_i$, the neighbor set $\mathcal{N}_j$ can be made available to node i via communication. Thus, by comparing these sets, it is easy for node i to know whether it belongs to a clique of order 5, without needing centralized information.

For the sake of simplicity, we will assume that all scarcely connected nodes are detected a priori following Remark 1 and removed from the network. Hence, we assume that each of all the (remaining) $n-4$ free nodes belong to at least one clique of order 5. However, belonging to a clique of order at least 5 is a necessary condition for localizability, but is not sufficient. Hence, we need to determine a sufficient condition for localizability, and then design an algorithm to classify the free nodes depending on their localizability, before any localization algorithm can be applied.

B. Problem Statement

In view of our problem setting, the distributed cooperative localization problem explored in this paper is described as follows. We consider a sensor network $\mathcal{G}$ with n nodes in a three-dimensional space, in which sensors can make range measurements and communicate with their neighbors on $\mathcal{G}$. We assume that the absolute positions of four anchor nodes $\mathbf{p_a}$ are known. Our objective is twofold.

Problem 1 (Localizability verification): Design a distributed localizability test scheme to determine for which of the free nodes it is possible to determine the absolute position.

Problem 2 (Localization algorithm): Design a localization algorithm to estimate $\mathbf{p_i}$ in finite time, for each free node $i \in \mathcal{V}_f$ that is localizable.

It is worth highlighting that the two problems described above are inherently interconnected. Specifically, the performance and accuracy of the localization algorithm (Problem 2) critically depend on the results obtained from the localizability verification step (Problem 1). Without prior verification of node localizability, the localization algorithm may mistakenly incorporate unlocalizable nodes, causing significant estimation errors due to error propagation during iterative computations. Such nodes not only fail to determine their own positions correctly, but also adversely affect the localization accuracy of theoretically localizable nodes. Therefore, performing localizability verification prior to localization is crucial for ensuring accurate and reliable positioning results across the entire network.

IV. Localizability Verification Algorithm

In this section, we address Problem 1. We start by providing a necessary and sufficient condition for node localizability. Then, we use such condition to build an algorithm to verify node localizability in a distributed fashion.

A. Necessary and Sufficient Node Localizability Condition

Before presenting the necessary and sufficient condition of node localizability, we first illustrate how the problem of estimating the Euclidean coordinates of a sensor network can be cast as a set of linear equations that can be derived in a distributed fashion using barycentric coordinates.

Consider a clique of order 5 formed by nodes $i,j,k,l,h \in \mathcal{V}$, with positions $\mathbf{p_i}, \mathbf{p_j}, \mathbf{p_k}, \mathbf{p_l}, \mathbf{p_h} \in \mathbb{R}^3$ in a framework we denote as $\mathcal{F}_p$. Node i can communicate with the other four neighbors and can get access to the range measurements d_{ij}, d_{ik}, d_{il}, and d_{ih} through onboard sensing or communication from its neighbors. Moreover, since communication and sensing networks coincide, node i can also receive direct information about measurements $d_{jk}, d_{jl}, d_{jh}, d_{kl}, d_{kh}$, and d_{lh} through communication with nodes j,k,l, and h. These measurements can be utilized to construct a matrix $D \in \mathbb{R}^{5\times 5}$ with its generic entry $D_{xy} := d_{xy}^2$, that is, equal to the square of the distance measured between nodes x and y for $x,y \in \{i,j,k,l,h\}$, and we define $J = I_5 - \frac{1}{5}\mathbf{1}_5\mathbf{1}_5^\top$. Then we can apply the Congruent Framework Construction (CFC)

Algorithm 1 CFC Algorithm

- **Function:** $Q = \mathbf{CFC}(D)$
1: Compute $X = -\frac{1}{2}JDJ$.
2: Compute singular value decomposition on X as $X = U\Lambda V^\top$, where $U = [\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{u_4}, \mathbf{u_5}]$ and $V = [\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}, \mathbf{v_4}, \mathbf{v_5}]$ are 5×5 unitary matrices and Λ is a diagonal matrix whose diagonal elements $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \lambda_5 \geq 0$ are singular values.
3: Compute $\Lambda_* = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) \in \mathbb{R}^{3\times 3}$ and $U_* = [\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}] \in \mathbb{R}^{5\times 3}$.
4: Compute $Q = \Lambda_*^{1/2} U_*^\top$.
5: **return** Q

algorithm summarized in the pseudocode in Algorithm 1, obtaining matrix $Q = [\mathbf{q_i}\, \mathbf{q_j}\, \mathbf{q_k}\, \mathbf{q_l}\, \mathbf{q_h}]$, to construct a congruent framework $\mathcal{F}_q$ for the clique.

Once a congruent framework of the clique of order 5 is generated, we compute the volumes of the tetrahedrons in the congruent framework using (3), and then obtain the barycentric coordinates $\{a_{ij}, a_{ik}, a_{il}, a_{ih}\}$ of node i using (2). Ultimately, using (1), we establish the following linear equation for the Euclidean coordinates of the five nodes belonging to the clique:

$$\mathbf{p_i} = a_{ij}\mathbf{p_j} + a_{ik}\mathbf{p_k} + a_{il}\mathbf{p_l} + a_{ih}\mathbf{p_h}, \tag{4}$$

where the constants a_{ij}, a_{ik}, a_{il}, and a_{ih} are known, and $\mathbf{p_i}, \mathbf{p_j}, \mathbf{p_k}, \mathbf{p_l}, \mathbf{p_h}$ are the unknown absolute Euclidean coordinates of nodes i, j, k, l, and h in the global coordinate system.

It is easy to find that a node may lie in multiple cliques. For any of them, we can establish one linear equality. Supposing that node i lies in a total of r_i cliques, we aggregate all the equations (4) for the free nodes in matrix form as

$$\left[\underbrace{\mathbf{p_1}\cdots\mathbf{p_1}}_{r_1}\cdots\underbrace{\mathbf{p_{n-4}}\cdots\mathbf{p_{n-4}}}_{r_{n-4}}\right]^\top = \left(\begin{bmatrix} C & B \end{bmatrix} \otimes I_3\right)\begin{bmatrix}\mathbf{p_f}\\ \mathbf{p_a}\end{bmatrix}, \tag{5}$$

where $C \in \mathbb{R}^{\left(\sum_{i=1}^{n-4} r_i\right)\times(n-4)}$ and $B \in \mathbb{R}^{\left(\sum_{i=1}^{n-4} r_i\right)\times 4}$ are two matrices that collects all the barycentric coordinates of the vertices associated with free nodes and anchor nodes, respectively. Define

$$E := [\underbrace{\mathbf{e_1}, \cdots, \mathbf{e_1}}_{r_1}, \underbrace{\mathbf{e_2}, \cdots, \mathbf{e_2}}_{r_2}, \cdots, \underbrace{\mathbf{e_{n-4}}, \cdots, \mathbf{e_{n-4}}}_{r_{n-4}}]^\top, \tag{6}$$

as a $\left(\sum_{i=1}^{n-4} r_i\right)\times(n-4)$ matrix. Let $\bar{C} = C\otimes I_3$, $\bar{B} = B\otimes I_3$ and $\bar{E} = E \otimes I_3$, we can write (5) in the following compact matrix form

$$\left(\bar{E} - \bar{C}\right)\mathbf{p_f} = \bar{B}\mathbf{p_a}. \tag{7}$$

Define $M = E - C$ and $\bar{M} = \bar{E} - \bar{C}$. Then (7) becomes

$$\bar{M}\mathbf{p_f} = \bar{B}\mathbf{p_a}, \tag{8}$$

where $\bar{M} \in \mathbb{R}^{\left(3\sum_{i=1}^{n-4} r_i\right)\times 3(n-4)}$.

Next, given the linear equation constraints in (8), we address the node localizability problem and present the necessary and sufficient conditions of node localizability.

The least squares method is considered to obtain the solution by minimizing the sum of the squares of the residuals made in the results of each individual equation. For the equation system (8), the least squares formula is obtained from the problem

$$\min_{\mathbf{p_f}} \|\bar{M}\mathbf{p_f} - \bar{B}\mathbf{p_a}\|, \tag{9}$$

the solution of which can be written with the normal equation

$$\mathbf{p_f} = \left(\bar{M}^\top\bar{M}\right)^{-1}\bar{M}^\top\bar{B}\mathbf{p_a}, \tag{10}$$

provided $\left(\bar{M}^\top\bar{M}\right)^{-1}$ exists, which is equivalent to $\bar{M}$ having full column rank. Note that for (10), an approximate solution is found when no exact solution exists. Moreover, (10) can be viewed as the solution to the following linear equation system

$$\bar{M}^\top\bar{M}\mathbf{p_f} = \bar{M}^\top\bar{B}\mathbf{p_a}, \tag{11}$$

if $\bar{M}^\top\bar{M}$ is invertible.

Remark 2: It is easy to see that $\mathbf{p_f}$ in (11) can be uniquely solved iff $\mathrm{rank}(\bar{M}^\top\bar{M}) = 3\,(n-4)$, which is equivalent to $\mathrm{rank}(\bar{M}) = 3\,(n-4)$ and $\mathrm{rank}(M) = n-4$. Hence if the rank of the matrix $\bar{M}$ is less than $3(n-4)$, namely, the rank of the matrix M is less than $n-4$, there must exist unlocalizable nodes in the sensor network.

By utilizing the definitions $\bar{M} = M \otimes I_3$ and $\bar{B} = B \otimes I_3$, we can interpret the linear equation system (11) as three equivalent linear equation systems: $M^\top M\mathbf{p_f^\varphi} = M^\top B\mathbf{p_a^\varphi}$, where the superscript $\varphi \in \{x, y, z\}$ denotes the Euclidean coordinate of φ axis. Here, $\mathbf{p_f^\varphi}$ is a vector comprising the φ axis Euclidean coordinates of all free nodes. Consequently, the localizability test for node i can be transformed into a more concise exploration based on the aforementioned linear equations for $\varphi \in \{x, y, z\}$, which share the same matrix $M^\top M$. This dimension reduction also leads to a reduction in computational complexity during the verification test. Therefore, we can conclude that a node $i \in \mathcal{V}_f$ is localizable if it belongs to a clique of order 5 and for any $\mathbf{x}^\varphi$ satisfying $M^\top M\mathbf{x}^\varphi = M^\top B\mathbf{p_a^\varphi}$, it must hold that $x_i^\varphi = p_i^\varphi$, where x_i^φ and p_i^φ are the corresponding scalars in $\mathbf{x_i} = [x_i^x, x_i^y, x_i^z]^\top$ and $\mathbf{p_i} = [p_i^x, p_i^y, p_i^z]^\top$, respectively.

Now we are ready to establish the necessary and sufficient conditions for localizability using the barycentric coordinates.

Theorem 1: Supposing that node i belongs to at least one clique of order 5, then node i is localizable iff $\mathbf{e_i} \perp \ker\left(M^\top M\right)$, where $\ker(M^\top M)$ denotes the kernel of the matrix $M^\top M$.

Proof: We start by proving sufficiency. Suppose to the contrary that node i is not localizable, that is to say, there exists a vector $\mathbf{x}^\varphi$ that satisfies $M^\top M\mathbf{x}^\varphi = M^\top B\mathbf{p_a^\varphi}$ but does not satisfy $x_i^\varphi = p_i^\varphi$. Then it can be obtained that $M^\top M(\mathbf{x}^\varphi - \mathbf{p_f^\varphi}) = 0$. That is, $\mathbf{x}^\varphi - \mathbf{p_f^\varphi} \in \mathrm{Ker}(M^\top M)$ but the ith component is nonzero. Hence, $\mathbf{e_i}^\top(\mathbf{x}^\varphi - \mathbf{p_f^\varphi}) \neq 0$, which contradicts to the condition $\mathbf{e_i}^\top \perp \ker(M^\top M)$.

Now, we prove necessity. If node i is localizable, then for any vectors $\mathbf{x}^\varphi$ and $\mathbf{y}^\varphi$ satisfying

$$M^\top M\mathbf{x}^\varphi = M^\top B\mathbf{p_a^\varphi}, \quad M^\top M\mathbf{y}^\varphi = M^\top B\mathbf{p_a^\varphi}, \tag{12}$$

it must hold that $x_i^\varphi = y_i^\varphi$, where x_i^φ and y_i^φ are the ith components of $\mathbf{x}^\varphi$ and $\mathbf{y}^\varphi$. That is to say, for any $w \in \ker(M^\top M)$, the ith component of w must be zero as otherwise there must exist two different vectors $\mathbf{x}^\varphi$ and $\mathbf{y}^\varphi$ satisfying (12) such that $x_i^\varphi \neq y_i^\varphi$. Therefore, $\mathbf{e_i} \perp \ker\left(M^\top M\right)$. ■

B. Distributed Verification Algorithm for Node Localizability

Building on Theorem 1, we design a distributed algorithm to address Problem 1, determining whether a generic node in the sensor network is localizable. In this algorithm, we compute the eigenvalues and eigenvectors of the matrix $M^\top M$ in a distributed fashion, following three steps. First, we design a subalgorithm to compute the local sum for each node based on the barycentric coordinates only through neighbor sensing and communication. Second, we propose a subalgorithm that achieves sum consensus among all nodes in the sensor network within a finite number of iterations. Third, leveraging the aforementioned algorithms, we design our distributed verification algorithm that solves Problem 1.

1) Local sum: Given an arbitrary matrix X with appropriate dimensions, where the ith row vector or column vector is denoted as $\mathbf{x_i}$, this Local Sum (LS) algorithm is designed to compute $\mathbf{y_i}$, the corresponding row or column vectors of the matrix products $M^\top MX$, $\bar{M}^\top \bar{M}X$, or $\bar{M}^\top \bar{B}X$, in a distributed manner. A pseudocode for this algorithm is reported in Subalgorithm 2.

In order to provide a clearer explanation of the algorithm, we recall that the matrix M is structured in such a way that each row consists of the barycentric coordinates a_{ij} of free nodes, i.e. $j \in \mathcal{N}_i \cap \mathcal{V}_f$, with the convention that $a_{ii} = -1$, and the remaining elements are zeros. While matrix B only comprises the barycentric coordinates a_{ij} of anchor nodes, i.e. $j \in \mathcal{N}_i \cap \mathcal{V}_a$. Note that the barycentric coordinates a_{ij} can be computed using the congruent framework by Algorithm 1. Exploiting these structures, the computation of each row in the matrix multiplication involving matrix M, $\bar{M}$, and $\bar{B}$ can be reformulated as a calculation of sums obtained by multiplying each a_{ij} with the corresponding vector $\mathbf{x_i}$ held by neighboring nodes. This is possible due to the fact that the elements in each row of M, $\bar{M}$, and $\bar{B}$ only involve the corresponding node and its neighbors. Hence we name this computation as Local Sum and it can be efficiently performed by enabling communication solely between nodes and their respective neighbors.

In this way, for each node i that has the knowledge of $\mathbf{x_i}$, after obtaining $\mathbf{x_j}$ by communicating with neighboring nodes, the barycentric coordinates a_{ij} can be utilized to calculate $\mathbf{y_i}$ using the LS algorithm in Subalgorithm 2. Specifically, each invocation of the LS Algorithm consists of two rounds of neighbor exchanges. In the first round, each node i collects the relevant vectors $\mathbf{x_j}$ from its neighbors and computes the intermediate quantities $\xi_\mathbf{i}^\mathbf{r}$. In the second round, each node i receives $a_{ji}^r \xi_\mathbf{j}^\mathbf{r}$ from its neighbors and finalizes the computation of its own output $\mathbf{y_i}$. This finite communication structure arises from the local structure of matrices $M, \bar{M}$, and $\bar{B}$, where each row explicitly involves only local barycentric coordinates. Furthermore, the computation of barycentric coordinates relies solely on local neighbor interactions, as detailed in Subsection IV-A. When all required distances are available within the network, these coordinates can be precomputed and stored in a distributed manner, eliminating the need for repeated calculations. Consequently, the LS Algorithm requires just two communication rounds and local computations, making it highly efficient.

On the one hand, the LS algorithm will be applied multiple times in the Distributed Localizability Verification Algorithm. Given the $(n-4)$-dimensional row vector $\mathbf{x_j}$ held by neighboring nodes of node i, it enables the local acquisition of $\mathbf{y_i}$ (the ith row vector of $M^\top MX$) through communication with neighbors. Here, X is an arbitrary $(n-4)\times(n-4)$ matrix, with $\mathbf{x_j}$ as the jth row vector. The whole process could be described as follows. After receiving $\mathbf{x_j}$ through communication with neighbors, the algorithm computes an intermediate result, consisting of the row vectors of $\xi = MX$. Then, employing another round of communication through which nodes exchange this intermediate result, nodes are finally able to compute the desired ith row vector of $M^\top MX$, here denoted by $\mathbf{y_i}$. This step is key for the distributed verification of the relation (11), as illustrated at the end of this section.

On the other hand, the LS Algorithm will also be used later in this paper to pursue localization. In that application, the input $\mathbf{x_j}$ is a 3×1 column vector denoting the coordinate estimate of node j. As a result, both $\xi_\mathbf{i}^\mathbf{r}$ and $\mathbf{y_i}$ are 3×1 column vectors. The LS Algorithm is then used to compute the components of $\mathbf{y} = \bar{M}^\top \bar{M}X$ or $\mathbf{y} = \bar{M}^\top \bar{B}X$, where X is an arbitrary $3(n-4)\times 1$ column vector consisting of several 3×1 vectors as its components, with the jth 3×1 vector as $\mathbf{x_j}$. Since $\bar{M}$ is composed of barycentric coordinates of free nodes, and $\bar{B}$ consists of barycentric coordinates of anchor nodes, we can differentiate the computation of $\bar{M}X$ or $\bar{B}X$ by selecting only free nodes or anchor nodes during the calculation of ξ in line 2. More specifically, we compute the components of $\xi = \bar{M}X$ when $j \in \{\mathcal{N}_i \cup i\} \cap \mathcal{V}_f \cap \triangle_i^r$ and $\xi = \bar{B}X$ when $j \in \{\mathcal{N}_i \cup i\} \cap \mathcal{V}_a \cap \triangle_i^r$, which will be specially noted when calling this subalgorithm. Note that the superscript r indicates that node i belongs to rth clique of order 5, more details can be found in Section IV-A.

Subalgorithm 2 LS Algorithm

- **Function:** $\mathbf{y_i} = \mathbf{LS}(\mathbf{x_j}, j \in \mathcal{N}_i)$

1: Receive $\mathbf{x_j}$ from its neighbor $j \in \mathcal{N}_i$.
2: Compute

$$\vdots$$
$$\xi_\mathbf{i}^\mathbf{r} = \sum_{j \in \{\mathcal{N}_i \cup i\} \cap \triangle_i^r} -a_{ij}^r \mathbf{x_j}$$
$$\vdots$$

3: Receive $a_{ji}^r \xi_\mathbf{j}^\mathbf{r}$ from its neighbor $j \in \mathcal{N}_i$.
4: Compute

$$\mathbf{y_i} = \sum_{j \in \{\mathcal{N}_i \cup i\} \cap \mathcal{V}_f, \forall r} -a_{ji}^r \xi_\mathbf{j}^\mathbf{r}.$$

5: **Return** $\mathbf{y_i}$.

2) Finite-Time K-Max-Consensus Sum: Next, we present the Finite-Time K-Max-Consensus Sum (FKMS) Algorithm, along with its associated function, the K-Max-Consensus (KMC) Algorithm. These algorithms are devised to achieve sum consensus among all nodes of the sensor network. Specifically, the algorithms enable each node to obtain the sum of all

the state of the nodes in the network in a distributed fashion through communication. In these two algorithms, ID_i and x_i denote the identifier and state value of node i, and δ denotes the network diameter, which we assumed to be known.

Remark 3: If the diameter δ is unknown, we can calculate it in a finite number of steps using established algorithms from the literature [41]–[43], which only require an upper bound on the number of nodes that can also be computed in a distributed fashion (see, e.g., [44], [45]).

The KMC algorithm, described in the pseudocode in Subalgorithm 3, is utilized to obtain the K maximum state values among all nodes in the sensor network and their corresponding identifiers. For each node, the entire process involves continuous comparison between its own K maximum state values and those held by its neighbors, ultimately obtaining the K maximum state values and corresponding identifiers. The key lies in the $\mathrm{K}_{\max}$ function, which sorts all state values of the node and its neighbors by value first and then by identifier if there are repeated values. Following this procedure, since the network is connected, each node in the sensor network will finally select the same K maximum values and corresponding identifiers. This iterative process requires δ steps, and the resulting values are stored in a local variable for each node, denoted by $\bar{\Omega}_i$, $i \in \mathcal{V}$.

Subalgorithm 3 KMC Algorithm

- **Function:** $\bar{\Omega}_i = \mathbf{KMC}\left(\left(ID_i, x_i\right), \delta\right)$
- **Initialization:**

1: $\Omega_i(0) = \{(ID_i, x_i)\}$.

- **Iteration:**

1: **for** $k = 1, k \leq \delta, k++$ **do**
2: Receive $\Omega_j(k-1)$ from its neighbor $j \in \mathcal{N}_i$.
3: $\Omega_i(k) = \mathbf{K}_{\max} \cup_{j \in \{\mathcal{N}_i \cup i\}} \Omega_j(k-1)$.
4: **end for**
5: **Return** $\bar{\Omega}_i = \Omega_i(k)$.

Then, the FKMS Algorithm (whose pseudocode is reported in Subalgorithm 4) uses the KMC Algorithm to enable each node to obtain the sum of all the state values held by all nodes. The overall process is described in the following three core steps of each iteration. First, each node updates its sum value by adding the K maximum values. Second, any state value that has already been included in the sum is assigned a value equal to $-\infty$. Third, the KMC Algorithm is executed to identify the subsequent set of K maximum values. These three steps are repeated iteratively until all state values eventually become equal to $-\infty$. This iterative process guarantees that each node eventually obtains the same sum value, which is referred to as sum consensus in this paper.

In Subalgorithm 4, x_i^{tmp} and sum_i denote the temporary variables of state value and sum of node i, respectively. The functions Keys$(\cdot)$ and Values$(\cdot)$ refer to the sets of identifiers and their corresponding state values. The final consensus sum value is denoted by $s\bar{u}m$. We initialize $\bar{\Omega} = \{(ID_i, x_i)\}$ and $x_i^{tmp} = x_i$ for each node. During the iteration, on line 2 we call the Subalgorithm 3 to get the K maximum state values and their identifiers for δ steps; then, line 3 updates the sum value by adding the nonnegative infinity values of the K maximum values once. In lines 4–5, we remove the nodes that have completed the accumulation of state values. At last, when all values $z \in$ Values $\left(\bar{\Omega}\right)$ are $-\infty$, the algorithm terminates and every node obtains the consensus sum result.

Subalgorithm 4 FKMS Algorithm

- **Function:** $s\bar{u}m = \mathbf{FKMS}\left(\left(ID_i, x_i\right), \delta, K\right)$
- **Initialization:**

1: $\bar{\Omega} = \{(ID_i, x_i)\}$.
2: $x_i^{tmp} = x_i$.
3: $sum_i = 0$.

- **Iteration:**

1: **while** $z \in$ Values $\left(\bar{\Omega}\right) > -\infty$ **do**
2: $\bar{\Omega} = \mathbf{KMC}\left(\left(ID_i, x_i^{tmp}\right), \delta\right)$.
3: $sum_i = sum_i + \sum_{z \in \text{Values}\left(\bar{\Omega}\right) \text{ and } z > -\infty} z$.
4: **if** $ID_i \in$ Keys $\left(\bar{\Omega}\right)$ or $x_i^{tmp} = -\infty$ **then**
5: $x_i^{tmp} = -\infty$.
6: **end if**
7: **end while**
8: **Return** $s\bar{u}m = sum_i$.

3) Distributed Localizability Verification: Finally, we present the Distributed Localizability Verification, whose goal is to check the localizability of each node. In accordance with Theorem 1, considering that eigenvectors corresponding to eigenvalues equal to 0 reside in the kernel set of a matrix, we address this localizability verification problem by conducting distributed computations of the matrix eigenvalues and eigenvectors of matrix $M^\top M$. Specifically, our algorithm, whose pseudocode is reported in Algorithm 5, computes for the eigenvalues of the matrix $M^\top M$ and calculates the ith component of each eigenvector at node i.

In Algorithm 5, the superscript $\top$ indicates it is a row vector and subscript s indicates the sth component of the vector, for example, $v_{i,s}^\top$ denotes the sth component of the row vector $\mathbf{v_i}^\top$. In the initialization, the initial value $\mathbf{v_i}^\top(0)$ of each node is set randomly. During the iteration, in line 1 we compute the components of $U = M^\top M V$ using the LS Algorithm, where matrix $V = [\mathbf{v_1}, \cdots, \mathbf{v_{n-4}}]^\top$ denotes the estimate of the matrix formed by the $n-4$ eigenvectors of $M^\top M$ as columns. Then, in line 2, we use the FKMS Algorithm to compute the $n-4$ eigenvalues using the Rayleigh quotient formula [46], [47]. This step allows each node to compute the eigenvalues through a distributed computation process. In line 3, we obtain the consensus matrix $O = U^\top U$ by computing all of its elements, using again the FKMS Algorithm. Then, in lines 4 and 5, we perform the distributed orthonormalization of U, i.e., $U^\top U = (VR)^\top (VR) = R^\top R$, where V and R are the QR factorization of U. Once the eigenvalues are deemed converged, each node i has access to $\mathbf{v_i}^\top$, which represents the ith component of all eigenvectors of $M^\top M$. This information is precisely what is needed to check the localizability verification condition for node i from Theorem 1. For practical applications, we set a maximum number of iterations $\mathrm{Iter}_{\max}$ and we reformulate the conditions as $\lambda_s(\mathrm{Iter}_{\max}) < \varepsilon_1$ and

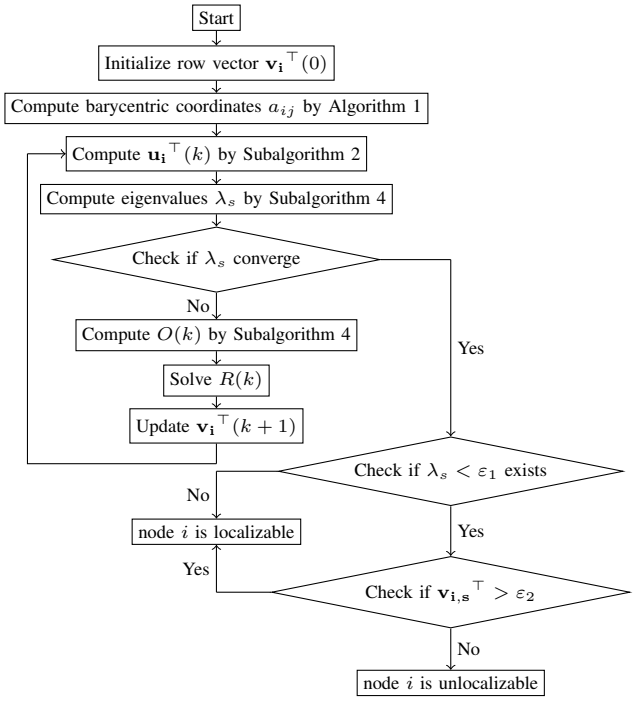

Fig. 2. Flow chart of the Distributed Localizability Verification Algorithm.

$\left|v_{i,s}^\top(\mathrm{Iter}_{\max})\right| > \varepsilon_2$, where $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$ are small threshold values. Fig. 2 reports a flow chart of the algorithm.

Algorithm 5 Distributed Localizability Verification Algorithm

- **Initialization:** $(\forall i = 1, \cdots, n-4)$:

1: $\mathbf{v_i}^\top(0) =$ random initial $(n-4)$ dimensional row vector.

- **Iteration:** with k starting at 0 :

1: $\mathbf{u_i}^\top(k) = \mathbf{LS}\left(\mathbf{v_j}^\top(k), j \in \{\mathcal{N}_i \cup i\} \cap \mathcal{V}_f\right)$.
2: $\lambda_s(k) = \frac{\mathbf{FKMS}\left(\left(ID_i, v_{i,s}^\top(k) u_{i,s}^\top(k)\right), \delta, K\right)}{\mathbf{FKMS}\left(\left(ID_i, v_{i,s}^\top(k) v_{i,s}^\top(k)\right), \delta, K\right)}, \forall s = 1, \cdots, n-4$.
3: $O(k) = \mathbf{FKMS}\left(\left(ID_i, \mathbf{u_i}(k)\mathbf{u_i}^\top(k)\right), \delta, K\right)$.
4: Solve $R(k)$ from $R^\top(k)R(k) = O(k)$.
5: $\mathbf{v_i}^\top(k+1) = \mathbf{u_i}^\top(k)R^{-1}(k)$.
6: $k = k+1$.

- **Localizability verification:**

1: **if** $\exists s$ such that $\lambda_s(\mathrm{Iter}_{\max}) < \varepsilon_1$ but $\left|v_{i,s}^\top(\mathrm{Iter}_{\max})\right| > \varepsilon_2$, **then**
2: node i is unlocalizable;
3: **else**
4: node i is localizable.
5: **end if**

Remark 4: We want to stress that the Distributed Localizability Verification Algorithm with the KMC and FKMS Algorithms can be used not only to solve Problem 1 in this setting, but their applicability can also be extended to all localization problems employing barycentric coordinates, namely, in two or three dimensional space with various types of measurements. For instance, it can be adopted before running the localization algorithms in [22]–[24] for the sensor network with unlocalizable nodes. The key lies in modifying the LS Algorithm according to the barycentric coordinates obtained in the specific scenario considered. Thus, our Distributed Localizability Verification Algorithm is a general distributed algorithm for testing the localizability of each node in the sensor network using barycentric coordinates.

V. Finite-Time Localization Algorithm

Here, we develop a finite-time distributed algorithm based on the conjugate gradient method to solve Problem 2 in finite time, and we analytically demonstrate its performance.

A. Distributed Localization Algorithm

Using the Distributed Localizability Verification Algorithm presented in Algorithm 5, each node is able to determine whether it is localizable or not. Then, if node i is not localizable, all cliques containing node i in the sensor network are removed, as well as the corresponding linear equations. At this stage, we define $\mathcal{V}_z$ as the set of n_z localizable free nodes, forming a sub-graph $\mathcal{G}_z = (\mathcal{V}_z, \mathcal{E}_z)$ with $\mathcal{E}_z := (i,j) \in \mathcal{E} : i,j \in \mathcal{V}_z$. Additionally, we make the following assumption.

Assumption 1: The sub-graph $\mathcal{G}_z = (\mathcal{V}_z, \mathcal{E}_z)$ is connected. Let δ_z be the diameter of this sub-graph. Under Assumption 1, we observe that the following inequalities naturally hold:

$$n_z \leq n-4 < n, \quad \delta_z \leq n_z - 1 < n. \tag{13}$$

The reduced set of linear equations from (11) that involve only nodes belonging to $\mathcal{V}_z$ can be written as follows:

$$\bar{M}_z^\top \bar{M}_z \mathbf{p_z} = \bar{M}_z^\top \bar{B}_z \mathbf{p_a}, \tag{14}$$

where the subscript z denotes the matrix or vector established by localizable nodes and $\mathbf{p_z}$ denotes the Euclidean coordinate vectors of the localizable node set $\mathcal{V}_z$.

Note that the matrix $\bar{M}_z^\top \bar{M}_z$ is symmetric and positive definite since all unlocalizable nodes are removed from the sensor network. Therefore, we propose our distributed localization algorithm to solve (14) by leveraging the conjugate gradient method. In our Distributed Localization Algorithm (whose pseudocode is reported in Algorithm 6), each node obtains an initial value $r_i(0)$, which is equal to the ith component of

$$\mathbf{r}(0) = \bar{M}_z^\top \bar{B}_z \mathbf{p_a} - \bar{M}_z^\top \bar{M}_z \hat{\mathbf{p}}_\mathbf{z}(0) = -\bar{M}_z^\top \left[\bar{M}_z - \bar{B}_z\right] \begin{bmatrix} \hat{\mathbf{p}}_\mathbf{z}(0) \\ \mathbf{p_a} \end{bmatrix}, \tag{15}$$

where $\hat{\mathbf{p}}_\mathbf{z}(0)$ denotes the initial estimate of the Euclidean coordinates of localizable nodes. During the iteration, in line 1 we use the LS Algorithm to compute $q_i(k)$ for node i, which is the ith components of $\mathbf{q}(k) = \bar{M}_z^\top \bar{M}_z \mathbf{v}(k)$. Then, in lines 2 and 5, we update the step sizes $\alpha_i(k)$ and $\beta_i(k)$. Note that $\alpha_i(k)$ and $\beta_i(k)$ are definitely constants consistent for all nodes, so they can be uniformly denoted as $\alpha(k)$ and $\beta(k)$ and computed as

$$\alpha(k) = \frac{\mathbf{r}^\top(k)\mathbf{r}(k)}{\mathbf{v}^\top(k)\mathbf{q}(k)}, \quad \beta(k) = \frac{\mathbf{r}^\top(k+1)\mathbf{r}(k+1)}{\mathbf{r}^\top(k)\mathbf{r}(k)}, \tag{16}$$

in a distributed fashion by means of the FKMS Algorithm. Meanwhile, in lines 3, 4, and 6, we update the estimate $\hat{\mathbf{p}}_\mathbf{i}(k)$ and parameters $\mathbf{r_i}(k)$ and $\mathbf{v_i}(k)$. Note that, the first two quantities are updated after computing the step $\alpha_i(k)$ since

they require the updated values of $\alpha_i(k)$, and before computing the step $\beta_i(k)$, for which they are needed. Similarly, the latter quantity ($\mathbf{v_i}(k)$) is updated at the end of the iteration, using the updated value of the step $\beta_i(k)$, computed in line 5.

Algorithm 6 Distributed Localization Algorithm

- **Initialization:** $(\forall i \in \mathcal{V}_z)$

1: $\hat{\mathbf{p}}_\mathbf{i}(0) =$ random initial 3 dimensional column vector.
2: $\hat{\mathbf{p}}_{\mathbf{n-3}}(0) = \mathbf{p_{n-3}}, \hat{\mathbf{p}}_{\mathbf{n-2}}(0) = \mathbf{p_{n-2}}, \hat{\mathbf{p}}_{\mathbf{n-1}}(0) = \mathbf{p_{n-1}}$ and $\hat{\mathbf{p}}_\mathbf{n}(0) = \mathbf{p_n}$.
3: $\mathbf{r_i}(0) = -\mathbf{LS}\left(\hat{\mathbf{p}}_\mathbf{j}(0), j \in \{\mathcal{N}_i \cup i\} \cap (\mathcal{V}_z \cup \mathcal{V}_a)\right)$.
4: $\mathbf{v_i}(0) = \mathbf{r_i}(0)$.

- **Iteration:** with k starting at 0

1: $\mathbf{q_i}(k) = \mathbf{LS}\ (\mathbf{v_j}(k), j \in \{\mathcal{N}_i \cup i\} \cap \mathcal{V}_z)$.
2: $\alpha_i(k) = \frac{\mathbf{FKMS}\left(\left(ID_i, \mathbf{r_i}^\top(k)\mathbf{r_i}(k)\right), \delta_z, K\right)}{\mathbf{FKMS}((ID_i, \mathbf{v_i}^\top(k)\mathbf{q_i}(k)), \delta_z, K)}$.
3: $\hat{\mathbf{p}}_\mathbf{i}(k+1) = \hat{\mathbf{p}}_\mathbf{i}(k) + \alpha_i(k)\mathbf{v_i}(k)$.
4: $\mathbf{r_i}(k+1) = \mathbf{r_i}(k) - \alpha_i(k)\mathbf{q_i}(k)$.
5: $\beta_i(k) = \frac{\mathbf{FKMS}\left(\left(ID_i, \mathbf{r_i}^\top(k+1)\mathbf{r_i}(k+1)\right), \delta_z, K\right)}{\mathbf{FKMS}((ID_i, \mathbf{r_i}^\top(k)\mathbf{r_i}(k)), \delta_z, K)}$.
6: $\mathbf{v_i}(k+1) = \mathbf{r_i}(k+1) + \beta_i(k)\mathbf{v_i}(k)$.
7: $k = k+1$.

B. Finite-Time Convergence Analysis

We prove that the Distributed Localization Algorithm in Algorithm 6 solves Problem 2, i.e., that the positions estimated by the algorithm converge to the real positions in a finite number of steps. The analysis consists of two parts. First, we prove that the FKMS Algorithm can obtain the step sizes α and β in finite time. Second, using this result, we prove finite-time convergence of Algorithm 6.

1) Finite-time computation of the step size: We start by proving the following lemma.

Lemma 1: Under Assumption 1, for any $i \in \mathcal{V}_z$ in the sensor network, $s\bar{u}m_i$ generated by the FKMS Algorithm in Subalgorithm 4 converges to the sum of the state values in finite time. Precisely, the algorithm takes no more than $\delta_z\left(\left\lceil \frac{n_z}{K} \right\rceil + 1\right)$ steps to converge.

Proof: Since the sub-graph is connected by Assumption 1, it is easy to observe that the KMC Algorithm succeeds within δ_z steps during the first iteration and the K maximum initial state values and their corresponding identifiers are obtained by all nodes. We denote them as $\bar{\Omega} = \mathbf{KMC}\left((ID_i, x_i(0)), \delta_z\right)$. Note that if state values are repeated, the largest identifiers are selected. After adding the K maximum initial state values to sum_i, the algorithm sets the temporary state values x_i^{tmp} of all nodes whose identifier is contained in Keys $\left(\bar{\Omega}\right)$ to $-\infty$, aiming to eliminate their value from subsequent KMC computations. Similar, in the following iterations, the sets $\bar{\Omega} = \mathbf{KMC}\left(\left(ID_i, x_i^{tmp}\right), \delta_z\right)$ are computed within δ_z steps each, where Values $\left(\bar{\Omega}\right)$ are the K maximum values of the initial state values except previous sets of K maximum values and Keys $\left(\bar{\Omega}\right)$ are their identifiers.

Hence, in each iteration, each node adds K maximum initial state values to sum_i. Once all initial state values have been accumulated, we have $x_i^{tmp} = -\infty$ for all nodes and Values $\left(\bar{\Omega}\right) = \{-\infty, \cdots, -\infty\}$. At this time, the algorithm terminates simultaneously for all nodes. This requires a number of iterations equal to the smallest integer that is larger than or equal to $\frac{n_z}{K}$. Furthermore, an additional step is needed to ensure that all the x_i^{tmp} values are equal to $-\infty$. Therefore, a total of $\left\lceil \frac{n_z}{K} \right\rceil + 1$ steps of the main routine of Subalgorithm 4 are required. Moreover, it should be noted that every node has the same value for sum_i at each step due to the consensus properties.

In conclusion, the KMC Algorithm requires δ_z steps for each iteration and the FKMS Algorithm needs to be executed no more than $\left\lceil \frac{n_z}{K} \right\rceil + 1$ times, yielding the claim. ■

In the light of Lemma 1, we can draw a conclusion that for any node $i \in \mathcal{V}_z$, the numerator and denominator of α_i and β_i can be calculated simultaneously in at most $\delta_z\left(\left\lceil \frac{n_z}{K} \right\rceil + 1\right)$ steps. Therefore, the step sizes α and β can be obtained in a finite number of steps, and the final results are shown as (16).

Remark 5: The parameter K in Subalgorithms 3 and 4 governs the tradeoff between memory consumption and iteration steps. Specifically, in the FKMS Algorithm, each node must store the top K state values along with their corresponding identifiers at each iteration, leading to a memory usage that grows linearly with K. However, the number of iterations required for convergence is inversely proportional to K, implying that a larger K reduces the total iteration count at the expense of increasing memory demand per node, and vice versa. Hence, in practical applications, K should be tuned based on the sensor hardware (e.g., available memory) and the required convergence speed. Furthermore, this tradeoff between memory and iteration steps is not exclusive to the FKMS Algorithm but is typical of distributed algorithms [48].

2) Finite-time convergence of Algorithm 6: After ensuring that the step size is obtained within a finite number of steps, we show that the Distributed Localization Algorithm converges in a finite number of iterations.

First, for all nodes i, $i \in \mathcal{V}_z$, the discrete-time system in Algorithm 6 can be written in the following compact matrix

$$\hat{\mathbf{p}}_\mathbf{z}(k+1) = \hat{\mathbf{p}}_\mathbf{z}(k) + \alpha(k)\mathbf{v}(k), \tag{17a}$$
$$\mathbf{r}(k+1) = \mathbf{r}(k) - \alpha(k)\bar{M}_z^\top \bar{M}_z \mathbf{v}(k), \tag{17b}$$
$$\mathbf{v}(k+1) = \mathbf{r}(k+1) + \beta(k)\mathbf{v}(k), \tag{17c}$$

with step sizes

$$\alpha(k) = \frac{\mathbf{r}^\top(k)\mathbf{r}(k)}{\mathbf{v}^\top(k)\bar{M}_z^\top \bar{M}_z \mathbf{v}(k)}, \tag{17d}$$
$$\beta(k) = \frac{\mathbf{r}^\top(k+1)\mathbf{r}(k+1)}{\mathbf{r}^\top(k)\mathbf{r}(k)}, \tag{17e}$$

by substituting $\mathbf{q}(k) = \bar{M}_z^\top \bar{M}_z \mathbf{v}(k)$ into (16). In the sequel, we denote $A = \bar{M}_z^\top \bar{M}_z$.

Then, before providing the proof, we present a lemma that demonstrates the orthogonality of the residuals and conjugacy of the search direction, considering [49].

Lemma 2: The residuals $\mathbf{r}(0), \mathbf{r}(1), \ldots$ generated by (17b) are mutually orthogonal and the direction vectors $\mathbf{v}(0), \mathbf{v}(1), \ldots$ generated by (17c) are mutually conjugate with respect to the matrix $A = \bar{M}_z^\top \bar{M}_z$.

Proof: The proof is reported in Appendix A. ■

Now, based on Lemma 2, we prove that the system (17) converges in m iterations, in the sense that the estimate $\hat{\mathbf{p}}_\mathbf{z}$ coincides with the exact solution $\mathbf{p_z}$ of the system of linear equations (14) after a finite number of iterations.

Lemma 3: For any $\hat{\mathbf{p}}_\mathbf{z}(0) \in \mathbb{R}^{3n_z}$, consider the sequence $\{\hat{\mathbf{p}}_\mathbf{z}(k)\}$, $k = 1, 2, \dots$ generated by (17). Then, there exists a constant $m \leq 3n_z$ such that $\hat{p}_z(k) = \mathbf{p_z}$, solution of the linear equation system (14), for any $k \geq m$.

Proof: Let m be the smallest integer such that the difference between $\hat{\mathbf{p}}_\mathbf{z}(0)$ and the solution $\mathbf{p_z}$ is in the subspace spanned by $\mathbf{v}(0), \mathbf{v}(1), \cdots, \mathbf{v}(m-1)$. Due to the fact that $A = \bar{M}_z^\top \bar{M}_z \in \mathbb{R}^{3n_z \times 3n_z}$ and $\hat{\mathbf{p}}_\mathbf{z} \in \mathbb{R}^{3n_z}$, the dimension of the subspace must be less than or equal to $3n_z$. Hence, $m \leq 3n_z$. Since the vectors $\mathbf{v}(0), \mathbf{v}(1), \cdots, \mathbf{v}(m-1)$ are linearly independent according to Lemma 2, we choose scalars $\mu(0), \cdots, \mu(m-1)$ such that we can write

$$\mathbf{p_z} = \hat{\mathbf{p}}_\mathbf{z}(0) + \mu(0)\mathbf{v}(0) + \cdots + \mu(m-1)\mathbf{v}(m-1). \quad (18)$$

Then, by substituting (18) into (15), we get

$$\mathbf{r}(0) = \mu(0)A\mathbf{v}(0) + \cdots + \mu(m-1)A\mathbf{v}(m-1), \quad (19)$$

with $b = \bar{M}_z^\top \bar{B}_z \mathbf{p_a}$. Next, using Lemma 2, we compute

$$\begin{aligned}\mathbf{v}^\top(k)\mathbf{r}(0) = \mathbf{v}^\top(k)\big(\mu(0)A\mathbf{v}(0) + \cdots + \mu(m-1)\cdot \\ A\mathbf{v}(m-1)\big) = \mu(k)\mathbf{v}^\top(k)A\mathbf{v}(k),\end{aligned} \quad (20)$$

and then obtain

$$\mu(k) = \frac{\mathbf{v}^\top(k)\mathbf{r}(0)}{\mathbf{v}^\top(k)A\mathbf{v}(k)} = \frac{\mathbf{r}^\top(k)\mathbf{r}(k)}{\mathbf{v}^\top(k)A\mathbf{v}(k)}. \quad (21)$$

So far, we find that $\alpha(k) = \mu(k)$, and hence that $\mathbf{p_z} = \hat{\mathbf{p}}_\mathbf{m}$, which yields the claim. ∎

Finally, we prove that Algorithm 6 solves Problem 2.

Theorem 2: For any $\hat{\mathbf{p}}_\mathbf{i}(0) \in \mathbb{R}^3$, $i \in \mathcal{V}_z$, the estimate $\hat{\mathbf{p}}_\mathbf{i}(k)$ generated by the Distributed Localization Algorithm in Algorithm 6 converges to the absolute position $\mathbf{p_i}$ in finite time. Specifically, it takes no more than $3n_z\left(2\delta_z\left(\lceil\frac{n_z}{K}\rceil + 1\right)\right)$ iterations to converge.

Proof: According to Lemma 1, the step sizes α_i and β_i can be obtained in at most $\delta_z\left(\lceil\frac{n_z}{K}\rceil + 1\right)$ iterations, Therefore, running the iteration in Algorithm 6 once requires a total of $2\delta_z\left(\lceil\frac{n_z}{K}\rceil + 1\right)$ iterations. Then, by applying the results of Lemma 3, we conclude that the bound on the convergence of Algorithm 6 is given by $3n_z\left(2\delta_z\left(\lceil\frac{n_z}{K}\rceil + 1\right)\right)$ iterations. ∎

Corollary 1: We can establish a conservative bound on the number of steps needed for convergence of the Distributed Localization Algorithm in terms of the total number of nodes of the network n, as being less than $3n\left(2n\left(\lceil\frac{n}{K}\rceil + 1\right)\right)$.

Proof: This comes directly from Theorem 2 and (13). ∎

Remark 6: Here, we provide a theoretical bound on the number of steps in Algorithm 6. However, in practical scenarios, due to the presence of the rounding errors, it is possible that a few additional steps may be needed.

Remark 7: We briefly analyze the communication overhead of the proposed distributed algorithms by considering the number of message exchanges required at each stage. The CFC Algorithm (Algorithm 1), which is executed only once, involves each clique of five nodes exchanging their pairwise squared distances to form matrix D, resulting in a total communication complexity of $O\left(|\mathcal{C}_5|\right)$, where $|\mathcal{C}_5|$ denotes the number of cliques of order 5 in $\mathcal{G}_z$. For the LS Algorithm (Subalgorithm 2), each invocation requires two rounds of neighbor exchanges, resulting in a complexity of $O\left(2\left|\mathcal{E}_z\right|\right)$, where $|\mathcal{E}_z|$ is the number of the edges in the network. The KMC Algorithm (Subalgorithm 3) executes δ_z iterations, exchanging messages of size proportional to K across edges, thus yielding a complexity of $O\left(\delta_z\left|\mathcal{E}_z\right|K\right)$. The FKMS Algorithm (Subalgorithm 4) builds on KMC and calls it $\lceil\frac{n_z}{K}\rceil + 1$ times to achieve sum consensus, resulting in complexity of $O\left(\delta_z\left|\mathcal{E}_z\right|K\left(\lceil\frac{n_z}{K}\rceil + 1\right)\right)$. Finally, each iteration of the Distributed Localization Algorithm (Algorithm 6) consists of one LS call and two FKMS calls, leading to a per-iteration complexity of $O\left(2\left|\mathcal{E}_z\right| + 2\delta_z\left|\mathcal{E}_z\right|K\left(\lceil\frac{n_z}{K}\rceil + 1\right)\right)$. As the localization algorithm converges within at most $3n_z$ iterations, the total communication overhead is upper-bounded by $O\left(3n_z\left[2\left|\mathcal{E}_z\right| + 2\delta_z\left|\mathcal{E}_z\right|K\left(\lceil\frac{n_z}{K}\rceil + 1\right)\right]\right)$, illustrating its dependence on the network topology, size, and parameter K.

VI. Simulations

To validate our algorithms and test their effectiveness, we present a set of numerical simulations performed on sensor networks of different size and in the presence of noise.

A. Case Study I

In the first case study, we consider a sensor network with $n = 50$ nodes consisting of four anchor nodes (selected at random) and 46 free nodes. Each node of the network is located in the three-dimensional space with each side of length 100m. Inspired by practical implementations [50]–[54], we establish that sensors can communicate and take exact measurements if they are within a certain radius. Here, we set such a critical radius at the value of 50m. Hence we generate the edges of the sensing and communication topology according to this rule, that is, $(i,j) \in \mathcal{E} \iff \|\mathbf{p_i} - \mathbf{p_j}\| \leq 50\text{m}$.

The configuration and the sensing and communication topology of the sensor network are shown in Fig. 3. Using the filtering process designed in Algorithm 5, we identified three unlocalizable nodes in the sensor network (see the red points in Fig. 3 and the blue circles in Fig. 4), which are excluded from the localization process.

Then, we run Algorithm 6 to estimate the positions of the remaining 43 localizable free nodes, starting from initial estimates picked uniformly at random in the domain $100\text{m} \times 100\text{m} \times 100\text{m}$. Fig. 4 illustrates the estimated positions at the end of the iterations (in red), i.e., after 5,369 steps (performed in approximately 18s on a 2.5 GHz 8-Core Intel Core i7-11700 computer), compared to the real positions (in blue). Consistent with our theoretical guarantees, the estimates for all localizable nodes converge to their real positions. We also illustrate a sample trajectory of coordinate estimates for a single node in Fig. 4. The trajectory shows how the estimate rapidly converges to the real position of the node, demonstrating the performance of our algorithm.

Now, we perform a comparison of our distributed localization algorithm (Algorithm 6) with the state-of-the-art

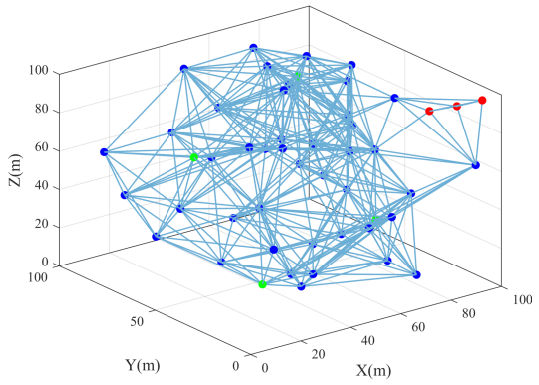

Fig. 3. Configuration and topology of the sensor network for Case Study I. Anchor nodes are denoted in green, unlocalizable nodes (detected by Algorithm 5) are in red, and localizable nodes in blue.

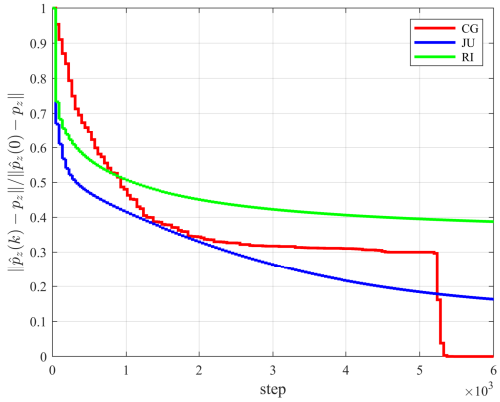

Fig. 5. Evolution of the estimation error ratio using Algorithm 6 (red), compared to JU [55] (blue) and RI algorithms [24] (green) for Case Study I.

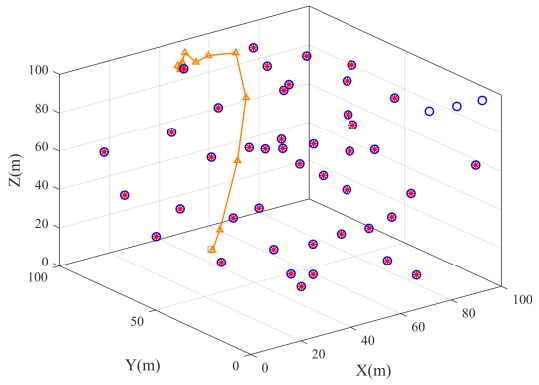

Fig. 4. Estimates obtained using the distributed localizability verification algorithm and our localization algorithm (red asterisks) compared to the real positions (blue circles) for Case Study I. The plot also depicts a sample trajectory: the initial estimate is represented by a grey square, the estimates at each 5 iterations by orange triangles.

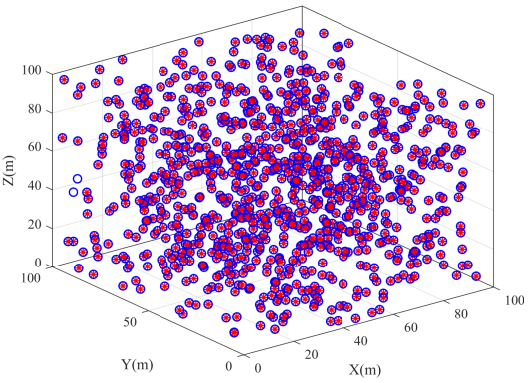

Fig. 6. Estimates obtained using the distributed localizability verification algorithm and our localization algorithm (red asterisks) compared to the real positions (blue circles) for Case Study II.

algorithms proposed in the literature. Specifically, we consider the Jacobi Under Relaxation Iteration algorithm (JU) from [55], which introduces a relaxation parameter to the Jacobi iterative estimation to enhance convergence speed, and the Richardson Iteration algorithm (RI), a widely used algorithm in the barycentric-coordinate cooperative localization problem, as proposed in [22]–[24]. In this comparison, we set the parameter of JU to $\alpha = 0.5$. While for RI, we use $\varepsilon = 2/\left(\min(\lambda\left(\bar{M}_z^\top \bar{M}_z\right)) + \max\left(\lambda(\bar{M}_z^\top \bar{M}_z)\right)\right)$ to achieve its maximum convergence speed. Additionally, we choose $K = 5$ in Subalgorithm 3 for our proposed method. It is worth noting that all three distributed localization algorithms are executed after excluding the unlocalizable nodes using Algorithm 5. In Fig. 5, we report the results of our comparison. In particular, the plot shows how the estimation error ratio $\|\hat{\mathbf{p}}_\mathbf{z}(k) - \mathbf{p}_\mathbf{z}\|/\|\hat{\mathbf{p}}_\mathbf{z}(0) - \mathbf{p}_\mathbf{z}\|$ obtained using our Algorithm 6 (denoted as CG) compared to the same quantity computed for the other two algorithms under the same settings (real positions and initial estimates). From this comparison, we note that, after a short transient, the distributed localization algorithm proposed in this paper outperforms the other two, making a significant enhancement in the convergence rate. Moreover, it eventually allows to determine the locations of all nodes in a finite number of steps as guaranteed by Theorem 2.

B. Case Study II

Then, we perform a set of simulations on a larger-scale network to verify the scalability of our algorithms. Specifically, we consider a scenario in which $n = 1,000$ sensors are placed in the same $100\text{m} \times 100\text{m} \times 100\text{m}$ three-dimensional space, with their positions set at random, and edges added if two sensors are within 20m of radius. Also in this case we assume that sensors perform exact measurements. As shown in Fig. 6, Algorithm 5 can successfully identify 2 unlocalizable nodes and, then, Algorithm 6 allows to determine the absolute positions of all the remaining 994 free nodes in finite time.

Similar to the previous case study, we compared the estimation error ratio of our algorithm with the other two algorithms from the state-of-the-art literature. The parameters for JU and RI remain unchanged from the previous case study. For our method, we choose $K = 20$ to balance computational efficiency and memory usage. The results, reported in Fig. 7,

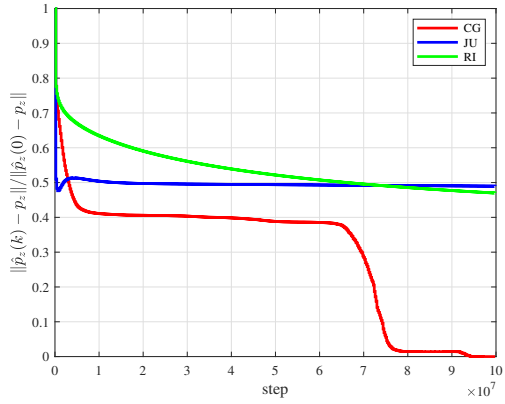

Fig. 7. Evolution of the estimation error ratio using Algorithm 6 (red), compared to JU [55] (blue) and RI algorithms [24] (green) for Case Study II.

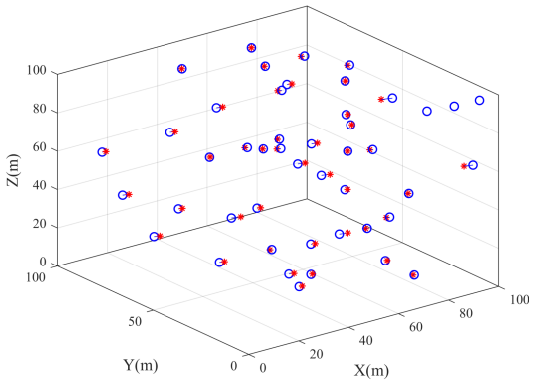

Fig. 8. Estimates obtained using the distributed localizability verification algorithm and our localization algorithm (red asterisks) compared to the real positions (blue circles) for Case Study III, with $\sigma = 0.01$.

show that as the number of nodes and network diameter increase, the performance of Algorithm 6 scales well. In fact, even though computational effort needed to estimate the positions increases, we observe that the convergence steps remain an order of magnitude smaller than the one needed for the other two algorithms. In addition, the number of steps can also be reduced at the cost of allocating more memory to the solver by increasing the value of the parameter K in the FKMS Algorithm, through the tradeoff between memory size and iteration steps discussed at the end of Section V.

C. Case Study III

Finally, we evaluate the performance of our proposed algorithms in the presence of noise by conducting a series of simulations using a network configuration identical to that of Case Study I, but assuming that range measurements are subject to noise. Specifically, we assume that, if $(i, j) \in \mathcal{E}$, then the range measurement that sensor i performs is equal to $d_{ij} = \hat{d}_{ij} + \Delta_{ij}$, where $\hat{d}_{ij}$ is the exact distance between sensors i and j, and Δ_{ij} is a Gaussian noise, with zero mean and standard deviation of $\sigma \geq 0$. It is important to notice that the noises Δ_{ij} and Δ_{ji} may not be equal, yielding $d_{ij} \neq d_{ji}$. However, we can address this discrepancy through a straightforward data pre-processing step to ensure symmetry of range measurements (i.e., $d_{ij} = d_{ji}$, if sensors i and j have the capability to sense and communicate with each other). The simplest approach to perform such data pre-processing involves to use a broadcast gossip-like consensus algorithm [56], so that range measures are averaged as $\bar{d}_{ij} = \bar{d}_{ji} = \frac{1}{2}(d_{ij} + d_{ji})$.

First, we consider a Gaussian noise with zero mean and standard deviation $\sigma = 0.01$ which applies to all the range measurements. By carrying out the simulation with Algorithm 5 and Algorithm 6, the corresponding results are presented in Fig. 8. The outcome indicates that, in the presence of noisy measurements, Algorithm 5 is still able to effectively identify unlocalizable nodes and, for each localizable free node, the estimated position (in red) ultimately converges in finite-time to a point closely approximating the true location (in blue) through Algorithm 6.

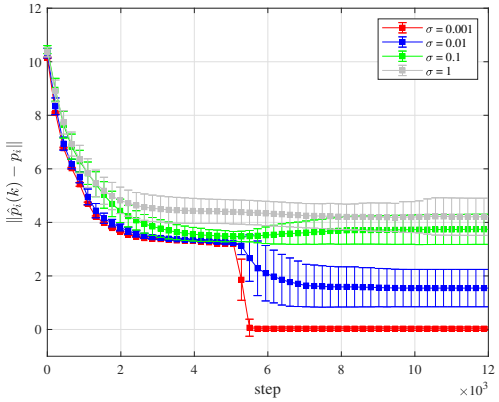

Fig. 9. Evolution of the mean estimation error using Algorithm 6 in Case Study III, with standard deviation of the noise $\sigma = 0.001$ (red), $\sigma = 0.01$ (blue), $\sigma = 0.1$ (green), and $\sigma = 1$ (gray). Vertical bars represent the standard deviation, computed over 100 independent runs of each scenario.

To further assess the impact of measurement noise on the performance of our proposed algorithms, we perform a set of simulations analyzing the temporal evolution of the mean absolute localization error for free nodes. Specifically, at each iteration k, the error is quantified as the mean Euclidean distance between the estimated and actual positions of all localizable free nodes. This analysis is performed for different values of the standard deviation of the measurement error σ. Due to the stochasticity of the error measurements, simulations are performed in a Monte Carlo fashion, by averaging each scenario of σ over 100 independent realizations of the stochastic measurements. The results, reported in Fig. 9, illustrate how our algorithm is able to deal with small-to-moderate levels of noise in the measures yielding finite-time convergence to a good proxy of the real locations. However, large values of noise hampers the localization algorithm, calling for the development of further tools to deal with these scenarios.

VII. Conclusion and Future Research

We studied the distributed localization problem for three-dimensional sensor networks with range measurements. First,

utilizing barycentric coordinates, we proposed a distributed localizability verification algorithm to identify which nodes are unlocalizable. Second, building on a conjugate gradient method, we proposed an efficient distributed localization algorithm, which is able to determine the location of all localizable nodes in finite time. Third, numerical simulations were offered to demonstrate the performance of our algorithm compared to the state-of-the-art, as well as its robustness in the presence of moderate noise in the range measurements.

The results presented in this paper pave the way for several avenues of future research. First, while our algorithms are developed in a theoretical framework, they offer fundamental insights that can inform practical localization applications. Potential applications include cooperative localization in GPS-denied environments, such as autonomous robotic networks and large-scale sensor deployments [57]–[59]. Future work should explore the integration of our methods into such technological systems, considering practical constraints such as hardware limitations and communication latency.

Second, real-world localization systems often have to deal with noisy and uncertain measurements. While our numerical simulations suggest that the algorithm is robust to moderate levels of noise in the measurements, performing a rigorous study of how noise propagates from the measurements to the estimates following, e.g., [60]–[62], is of paramount importance for assessing their applicability in real-world scenarios. This extension involves addressing critical challenges such as localization ambiguity arising from noisy measurements, node localizability conditions under uncertainty, and establishing convergence guarantees in noisy environments. Furthermore, factors such as anchor node placement and network density play a significant role in localization accuracy under noise. Future research should also consider spatially varying noise levels, as measurement uncertainty is typically non-uniform in large-scale networks.

Third, the reliance of distributed algorithms on communication introduces significant challenges related to communication overhead and energy efficiency, particularly in resource-constrained sensor networks. Future research should focus on systematically evaluating and optimizing these factors to improve the practicality, sustainability, and overall effectiveness of the proposed distributed localization method in real-world deployments.

Finally, an interesting future research direction is to improve the performance of our algorithm, designing simultaneous localizability verification and localization. Additionally, generalizing our algorithms to deal with dynamic networks, such as tracking scenarios, is also a topic that warrants consideration. Moreover, the methodologies presented in this paper have broader applicability beyond the specific localization problem addressed herein. In particular, the finite-time K-Max-Consensus Sum algorithm developed during this study can be effectively adapted to various problems requiring finite-time consensus computation, thereby expanding the practical utility and theoretical significance of our contributions.

APPENDIX

A. Proof of the Lemma 2

The proof of the lemma is equivalent to prove that the following relations hold

$$\mathbf{r}^\top(g)\mathbf{r}(f) = 0 \ (g \neq f), \tag{22a}$$

$$\mathbf{v}^\top(g)A\mathbf{v}(f) = 0 \ (g \neq f), \tag{22b}$$

$$\mathbf{v}^\top(g)\mathbf{r}(f) = 0 \ (g < f),\ \mathbf{v}^\top(g)\mathbf{r}(f) = \mathbf{r}^\top(g)\mathbf{r}(g), \tag{22c}$$

$$\mathbf{r}^\top(g)A\mathbf{v}(g) = \mathbf{v}^\top(g)A\mathbf{v}(g),\ \mathbf{r}^\top(g)A\mathbf{v}(f) = 0 \ (g \neq l, g \neq f+1), \tag{22d}$$

which could be made by induction.

First, it easy to get that the vectors $\mathbf{r}(0), \mathbf{v}(0)$ and $\mathbf{r}(1)$ satisfy relations (22) due to

$$\begin{aligned}\mathbf{r}^\top(0)\mathbf{r}(1) &= \mathbf{v}^\top(0)\mathbf{r}(1) = \mathbf{v}^\top(0)\left(\mathbf{r}(0) - \alpha(0)A\mathbf{v}(0)\right)\\ &= \mathbf{r}^\top(0)\mathbf{r}(0) - \alpha(0)\mathbf{v}^\top(0)A\mathbf{v}(0)\\ &= \mathbf{r}^\top(0)\mathbf{r}(0) - \frac{\mathbf{r}^\top(0)\mathbf{r}(0)}{\mathbf{v}^\top(0)A\mathbf{v}(0)}\left(\mathbf{v}^\top(0)A\mathbf{v}(0)\right) = 0,\end{aligned} \tag{23}$$

by (17b) and (17d).

Second, supposing that relations (22) hold for the vectors $\mathbf{r}(0), \cdots, \mathbf{r}(k)$ and $\mathbf{v}(0), \cdots, \mathbf{v}(k-1)$, we prove that $\mathbf{v}(k)$ satisfies relations (22). To verify that $\mathbf{v}(k)$ can be adjoined to this set, it is necessary to show that

$$\mathbf{r}^\top(g)\mathbf{v}(k) = \mathbf{r}^\top(k)\mathbf{r}(k) \quad (g \leq k), \tag{24a}$$

$$\mathbf{v}^\top(g)A\mathbf{v}(k) = 0 \quad (g < k), \tag{24b}$$

$$\mathbf{r}^\top(k)A\mathbf{v}(g) = \mathbf{v}^\top(k)A\mathbf{v}(g) \quad (g \leq k, g \neq k-1). \tag{24c}$$

It can be verified that (17e) and (17c) hold for any k iff

$$\mathbf{v}(k) = \left(\mathbf{r}^\top(k)\mathbf{r}(k)\right)\sum\nolimits_{f=0}^{k}\frac{\mathbf{r}(f)}{\mathbf{r}^\top(f)\mathbf{r}(f)}. \tag{25}$$

Then for formula 24, (24a) can be derived at once from (25) and (22a) as follows

$$\begin{aligned}\mathbf{r}^\top(g)\mathbf{v}(k) &= \mathbf{r}^\top(g)\left(\mathbf{r}^\top(k)\mathbf{r}(k)\sum\nolimits_{l=0}^{k}\frac{\mathbf{r}(f)}{\mathbf{r}^\top(f)\mathbf{r}(f)}\right)\\ &= \mathbf{r}^\top(g)\left(\mathbf{r}^\top(k)\mathbf{r}(k)\frac{\mathbf{r}(g)}{\mathbf{r}^\top(g)\mathbf{r}(g)}\right) = \mathbf{r}^\top(k)\mathbf{r}(k).\end{aligned} \tag{26}$$

To prove (24b), we use (17b) and find that

$$\begin{aligned}\mathbf{r}^\top(g+1)\mathbf{v}(k) &= \left(\mathbf{r}(g) - \alpha(g)A\mathbf{v}(g)\right)^\top\mathbf{v}(k)\\ &= \mathbf{r}^\top(g)\mathbf{v}(k) - \alpha(g)\mathbf{v}^\top(g)A^\top\mathbf{v}(k),\end{aligned} \tag{27}$$

which becomes

$$\mathbf{r}^\top(k)\mathbf{r}(k) = \mathbf{r}^\top(k)\mathbf{r}(k) - \alpha(g)\mathbf{v}^\top(g)A\mathbf{v}(k) \tag{28}$$

by (24a) when $g < k$, then (24b) holds since $\alpha(g) > 0$. In order to establish (24c), we use (17c) and (24b) to obtain

$$\begin{aligned}\mathbf{v}^\top(k)A\mathbf{v}(g) &= \left(\mathbf{r}(k) + \beta(k-1)\mathbf{v}(k-1)\right)^\top A\mathbf{v}(g)\\ &= \mathbf{r}^\top(k)A\mathbf{v}(g) + \beta(k-1)\mathbf{v}^\top(k-1)A\mathbf{v}(g)\\ &= \mathbf{r}^\top(k)A\mathbf{v}(g),\end{aligned} \tag{29}$$

when $g \neq k-1$, which follows that (24c) holds. Therefore, relations (22) holds for the vectors $\mathbf{r}(0), \mathbf{r}(1), \cdots, \mathbf{r}(k)$ and $\mathbf{v}(0), \mathbf{v}(1), \cdots, \mathbf{v}(k)$.

Finally, supposing that relations (22) holds for the vectors $\mathbf{r}(0), \cdots, \mathbf{r}(k)$ and $\mathbf{v}(0), \cdots, \mathbf{v}(k-1)$, we prove $\mathbf{r}(k+1)$

satisfies relations (22). With the above results, proving that $\mathbf{r}(k+1)$ can be adjoined to this set is done by showing that

$$\mathbf{r}^\top(g)\mathbf{r}(k+1)=0 \quad (g\le k), \tag{30a}$$
$$\mathbf{v}^\top(g)A^\top\mathbf{r}(k+1)=0 \quad (g< k), \tag{30b}$$
$$\mathbf{v}^\top(g)\mathbf{r}(k+1)=0 \quad (g\le k). \tag{30c}$$

By (17b), we have

$$\mathbf{r}^\top(g)\mathbf{r}(k+1)=\mathbf{r}^\top(g)\mathbf{r}(k)-\alpha(k)\mathbf{r}^\top(g)A\mathbf{v}(k). \tag{31}$$

When $g<k$, the terms on the right of (31) are both 0 and then (30a) holds. When $g=k$, the right member is still zero

$$\begin{aligned}&\mathbf{r}^\top(k)\mathbf{r}(k)-\alpha(k)\mathbf{r}^\top(k)A\mathbf{v}(k)\\&=\mathbf{r}^\top(k)\mathbf{r}(k)-\tfrac{\mathbf{r}^\top(k)\mathbf{r}(k)}{\mathbf{v}^\top(k)A\mathbf{v}(k)}\left(\mathbf{r}^\top(k)A\mathbf{v}(k)\right)\\&=\mathbf{r}^\top(k)\mathbf{r}(k)-\mathbf{r}^\top(k)\mathbf{r}(k)=0,\end{aligned} \tag{32}$$

by (17d) and (22d). Therefore, (30a) holds. Besides, when $g<k$, using (17b) again, we have

$$\begin{aligned}0=\mathbf{r}^\top(k+1)\mathbf{r}(g+1)=\mathbf{r}^\top(k+1)\mathbf{r}(g)-\\ \alpha(g)\mathbf{r}^\top(k+1)A\mathbf{v}(g)=-\alpha(g)\mathbf{r}^\top(k+1)A\mathbf{v}(g),\end{aligned} \tag{33}$$

hence (30b) holds. The equation (30c) follows from (30a) and the formula (25) for $v(g)$, if $g\le k$, we have

$$\mathbf{v}^\top(g)\mathbf{r}(k+1)=\left(\left(\mathbf{r}^\top(g)\mathbf{r}(g)\right)\sum_{f=0}^{g}\frac{\mathbf{r}(f)}{\mathbf{r}^\top(f)\mathbf{r}(f)}\right)^\top\mathbf{r}(k+1)=0. \tag{34}$$

Therefore, (22) hold for the vectors $\mathbf{r}(0),\mathbf{r}(1),\cdots,\mathbf{r}(k),\mathbf{r}(k+1)$ and $\mathbf{v}(0),\mathbf{v}(1),\cdots,\mathbf{v}(k)$. ■

REFERENCES

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, 2005.

[2] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 24–40, 2005.

[3] J. R. Lowell, "Military applications of localization, tracking, and targeting," *IEEE Wirel. Commun.*, vol. 18, no. 2, pp. 60–65, 2011.

[4] B. Bayat, N. Crasta, A. Crespi, A. M. Pascoal, and A. Ijspeert, "Environmental monitoring using autonomous vehicles: a survey of recent searching techniques," *Curr. Opin. Biotechnol.*, vol. 45, pp. 76–84, 2017.

[5] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2655–2669, 2017.

[6] T. Wang, A. Conti, and M. Z. Win, "Network navigation with scheduling: Distributed algorithms," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1319–1329, 2019.

[7] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex laplacian," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1765–1777, 2014.

[8] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Artech House, 2002.

[9] R. M. Buehrer, H. Wymeersch, and R. M. Vaghefi, "Collaborative sensor network localization: Algorithms and practical issues," *Proc. IEEE*, vol. 106, no. 6, pp. 1089–1114, 2018.

[10] C. Yang and H.-R. Shao, "WiFi-based indoor positioning," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 150–157, 2015.

[11] X. Tong, H. Li, X. Tian, and X. Wang, "Wi-Fi localization enabling self-calibration," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 904–917, 2021.

[12] H. Soganci, S. Gezici, and H. V. Poor, "Accurate positioning in ultra-wideband systems," *IEEE Wirel. Commun.*, vol. 18, no. 2, pp. 19–27, 2011.

[13] F. Shan, H. Huo, J. Zeng, Z. Li, W. Wu, and J. Luo, "Ultra-wideband swarm ranging protocol for dynamic and dense networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2834–2848, 2022.

[14] F. Gu, J. Niu, and L. Duan, "WAIPO: A fusion-based collaborative indoor localization system on smartphones," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2267–2280, 2017.

[15] B. Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM J. Optim.*, vol. 5, no. 4, pp. 835–857, 1995.

[16] J. Aspnes *et al.*, "A theory of network localization," *IEEE Trans. Mob. Comput.*, vol. 5, no. 12, pp. 1663–1678, 2006.

[17] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–53, 2007.

[18] G. Jing, C. Wan, and R. Dai, "Angle-based sensor network localization," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 840–855, 2021.

[19] U. Helmke and B. D. Anderson, "Equivariant morse theory and formation control," in *51st Annu. Allerton Conf. Commun. Control Comput.*, 2013, pp. 1576–1583.

[20] U. A. Khan, S. Kar, and J. M. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 2000–2016, 2009.

[21] Y. Diao, M. Fu, Z. Lin, and H. Zhang, "A sequential cluster-based approach to node localizability of sensor networks," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 4, pp. 358–369, 2015.

[22] Y. Diao, Z. Lin, and M. Fu, "A barycentric coordinate based distributed localization algorithm for sensor networks," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4760–4771, 2014.

[23] P. Cheng, T. Han, X. Zhang, R. Zheng, and Z. Lin, "A single-mobile-anchor based distributed localization scheme for sensor networks," in *35th Chin. Control Conf.*, 2016, pp. 8026–8031.

[24] T. Han, Z. Lin, R. Zheng, Z. Han, and H. Zhang, "A barycentric coordinate based approach to three-dimensional distributed localization for wireless sensor networks," in *13th IEEE Int. Conf. Control Autom.*, 2017, pp. 600–605.

[25] S. Zhao and D. Zelazo, "Localizability and distributed protocols for bearing-based network localization in arbitrary dimensions," *Automatica*, vol. 69, pp. 334–341, 2016.

[26] Z. Lin, T. Han, R. Zheng, and M. Fu, "Distributed localization for 2-D sensor networks with bearing-only measurements under switching topologies," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6345–6359, 2016.

[27] X. Li, X. Luo, and S. Zhao, "Globally convergent distributed network localization using locally measured bearings," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 1, pp. 245–253, 2019.

[28] K. Cao, Z. Han, Z. Lin, and L. Xie, "Bearing-only distributed localization: A unified barycentric approach," *Automatica*, vol. 133, p. 109834, 2021.

[29] P. Barooah and J. P. Hespanha, "Estimation on graphs from relative measurements," *IEEE Control Syst.*, vol. 27, no. 4, pp. 57–74, 2007.

[30] Z. Lin, M. Fu, and Y. Diao, "Distributed self localization for relative position sensing networks in 2-D space," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3751–3761, 2015.

[31] Z. Lin, Z. Han, and M. Cao, "Distributed localization for multi-robot systems in presence of unlocalizable robots," in *59th IEEE Conf. Decis. Control.*, 2020, pp. 1550–1555.

[32] L. Chen, "Triangular angle rigidity for distributed localization in 2D," *Automatica*, vol. 143, p. 110414, 2022.

[33] L. Chen, K. Cao, L. Xie, X. Li, and M. Feroskhan, "3-D network localization using angle measurements and reduced communication," *IEEE Trans. Signal Process.*, vol. 70, pp. 2402–2415, 2022.

[34] Z. Lin, T. Han, R. Zheng, and C. Yu, "Distributed localization with mixed measurements under switching topologies," *Automatica*, vol. 76, pp. 251–257, 2017.

[35] X. Fang, X. Li, and L. Xie, "3-D distributed localization with mixed local relative measurements," *IEEE Trans. Signal Process.*, vol. 68, pp. 5869–5881, 2020.

[36] A. F. Möbius, *Der Barycentrische Calcul*. Verlag von Johann Ambrosius Barth, 1827.

[37] H. Ping, Y. Wang, X. Shen, D. Li, and W. Chen, "On node localizability identification in barycentric linear localization," *ACM Trans. Sens. Netw.*, vol. 19, p. 19, 2022.

[38] H. Ping, Y. Wang, D. Li, and W. Chen, "Understanding node localizability in barycentric linear localization," *IEEE/ACM Trans. Netw.*, 2022.

[39] A. Singer and M. Cucuringu, "Uniqueness of low-rank matrix completion by rigidity theory," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 4, pp. 1621–1641, 2010.

[40] T. Eren *et al.*, "Rigidity, computation, and randomization in network localization," in *23rd IEEE Int. Conf. Comput. Commun.*, vol. 4, 2004, pp. 2673–2684.
[41] P. S. Almeida, C. Baquero, and A. Cunha, "Fast distributed computation of distances in networks," in *51st IEEE Conf. Decis. Control.*, 2012, pp. 5215–5220.
[42] D. Peleg, L. Roditty, and E. Tal, "Distributed algorithms for network diameter and girth," in *Automata, Languages, and Programming*, 2012, pp. 660–672.
[43] G. Oliva, R. Setola, and C. N. Hadjicostis, "Distributed finite-time calculation of node eccentricities, graph radius and graph diameter," *Syst. Control Lett.*, vol. 92, pp. 20–27, 2016.
[44] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed cardinality estimation in anonymous networks," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 645–659, 2013.
[45] S. Zhang, C. Tepedelenlioğlu, M. K. Banavar, and A. Spanias, "Distributed node counting in wireless sensor networks in the presence of communication noise," *IEEE Sens. J.*, vol. 17, no. 4, pp. 1175–86, 2016.
[46] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
[47] B. N. Parlett, *The Symmetric Eigenvalue Problem*. SIAM, 1998.
[48] L. Zino, B. Barzel, and A. Rizzo, "Network science and automation," in *Springer Handbook of Automation*, S. Y. Nof, Ed. Cham: Springer International Publishing, 2023, pp. 251–274.
[49] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand.*, vol. 49, no. 6, p. 409, 1952.
[50] D. Niculescu and B. Nath, "Ad Hoc positioning system (APS) using AOA," in *22nd IEEE Int. Conf. Comput. Commun.*, vol. 3, 2003, pp. 1734–1743.
[51] Y. Zhang, W. Liu, Y. Fang, and D. Wu, "Secure localization and authentication in ultra-wideband sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 4, pp. 829–835, 2006.
[52] J. Vetelino and A. Reghu, *Introduction to Sensors*. CRC Press, 2017.
[53] "IEEE standard for low-rate wireless networks–Amendment 1: Enhanced ultra wideband (UWB) physical layers (PHYs) and associated ranging techniques," *IEEE Std 802.15.4z-2020 (Amendment to IEEE Std 802.15.4-2020)*, pp. 1–174, 2020.
[54] P. Angueira, I. Val, J. Montalban, Ó. Seijo, E. Iradier, P. S. Fontaneda, L. Fanari, and A. Arriola, "A survey of physical layer techniques for secure wireless communications in industry," *IEEE Commun. Surv. Tut.*, vol. 24, no. 2, pp. 810–838, 2022.
[55] Y. Xia, C. Yu, and C. He, "An exploratory distributed localization algorithm based on 3D barycentric coordinates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 702–712, 2022.
[56] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2748–2761, 2009.
[57] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, 2011.
[58] F. Shan, H. Huo, J. Zeng, Z. Li, W. Wu, and J. Luo, "Ultra-wideband swarm ranging protocol for dynamic and dense networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 6, pp. 2834–2848, 2022.
[59] L. Chen, J. Xiao, R. C. H. Lin, and M. Feroskhan, "Angle-constrained formation maneuvering of unmanned aerial vehicles," *IEEE Trans. Control Syst. Technol*, vol. 31, no. 4, pp. 1733–1746, 2023.
[60] Q. Xiao, B. Xiao, K. Bu, and J. Cao, "Iterative localization of wireless sensor networks: an accurate and robust approach," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 608–621, 2013.
[61] B. Wang and Y.-P. Tian, "Distributed network localization: Accurate estimation with noisy measurement and communication information," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5927–5940, 2018.
[62] F. Xiao, L. Chen, C. Sha, L. Sun, R. Wang, A. X. Liu, and F. Ahmed, "Noise tolerant localization for sensor networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1701–1714, 2018.

Jinze Wu received her B.Eng. Degree in Vehicle Engineering from Dalian University of Technology, Dalian, China, in 2018, and her M.Eng. Degree in Aeronautical and Astronautical Science and Technology from Sun Yat-sen University, Shenzhen, China, in 2022. She is currently pursuing her PhD in the Department of Electronics and Telecommunications at Politecnico di Torino, Turin, Italy, and the School of Automation and Intelligent Manufacturing at Southern University of Science and Technology, Shenzhen, China. Her research interests mainly focus on multiagent systems and distributed control and algorithms.

Lorenzo Zino is an Assistant Professor with the Department of Electronics and Telecommunications, Politecnico di Torino, Italy, since 2022. He received BS (2012), MS (summa cum laude, 2014), and PhD (with honors, 2018) in Applied Mathematics from Politecnico di Torino. He held Research Fellowships at Politecnico di Torino, University of Groningen, and New York University Tandon School of Engineering. His research interests include modeling, analysis, and control of dynamics over networks, applied probability, and game theory. He has co-authored more than 80 international scientific publications, including 50 journal papers. In 2024, he was the recipient of the Best Young Author Journal Paper Award from the IEEE CSS Italy Chapter. He is member of the Editorial Board of *Scientific Reports*, Associate Editor of the *Journal of Computational Science*, and member of the CEB for IEEE CSS and EUCA.

Zhiyun Lin received the Ph.D. degree in electrical and computer engineering from University of Toronto, Canada, in 2005. He is currently a Full Professor with the Southern University of Science and Technology, China. Preceding to this position, he was a Postdoctoral Research Associate (2005–07) with the University of Toronto, then worked as a Research Professor (2007–11) and a Full Professor (2012–17) at Zhejiang University, China, and served as the Director of Artificial Intelligence Institute (2017–21), Hangzhou Dianzi University, China. He has held many visiting Professor appointments, including Yale University, the Australian National University, University of Groningen, University of Cagliari, University of Newcastle, University of Technology Sydney. His research interests focus on multiagent systems, distributed artificial intelligence, autonomous systems and swarm robots, and cyber-physical systems. He is an Associate Editor for IEEE Systems Journal. He is a fellow of IEEE and IET.

Alessandro Rizzo (Senior Member, IEEE) is an Associate Professor at the Department of Electronics and Telecommunications, Politecnico di Torino, Italy, where he directs the Complex Systems Laboratory. He received his Laurea degree in Computer Engineering, summa cum laude, and his PhD in Automation and Electronics Engineering from the University of Catania, Italy, in 1996 and 2000, respectively. Dr. Rizzo's previous affiliations include JET Joint Undertaking (UK), ST Microelectronics (Italy), the University of Messina (Italy), Politecnico di Bari (Italy), and New York University (USA). His research interests span complex networks and systems, robotics, and the modeling and control of nonlinear systems. He has authored two books, holds two international patents, and has published over 200 papers in peer-reviewed international journals and conference proceedings. A Distinguished Lecturer for the IEEE Nuclear and Plasma Sciences Society, he received the Best Application Paper Award at the 2002 IFAC World Congress, as well as two Amazon Research Awards in Robotics (2019 and 2021).