Sensor Network Localization via Alternating Rank Minimization Algorithms

Changhuang Wan, Gangshan Jing, Sixiong You, and Ran Dai

***Abstract*—Sensor network localization (SNL) is to determine physical coordinates of all sensors in a network given global coordinates of anchors and available measurements among sensors and anchors. Two challenges related to SNL are to find conditions leading to a uniquely localizable network and develop effective and efficient methods to solve SNL problems. This work first proves that infinitesimal rigidity, together with some mild conditions, is sufficient for unique localizability of a network considering additional relationships between nonadjacent sensors. On the other hand, solving an SNL problem is generally NP-hard due to its nonconvex constraints. Instead of ignoring the rank constraint used in existing relaxation methods, we convert the rank constraint in the SNL problem into its equivalent constraints and solve it alternatively by proposing the alternating rank minimization algorithm (ARMA). We start with the centralized ARMA to solve the exact SNL problem. Next, to improve the scalability for solving large-scale SNL problems, ARMA is extended in a distributed manner by decomposing the original problem into a group of subproblems, which can be solved independently. Finally, simulation cases are provided for both centralized and distributed ARMA to validate the improved localization accuracy, efficiency, and robustness by being compared to the state-of-the-art localization methods.**

***Index Terms*—Distributed optimization, graph rigidity, rank-constrained optimization, sensor network localization (SNL).**

I. Introduction

Wireless-sensor networks, due to their capabilities of sensing, processing, and communication, have a wide range of applications [1], such as target tracking and detection [2], [3]; process control; environment monitoring [4]–[6]; area exploration [7], [8]; data collection; and cooperative robots [9], [10], just to name a few. Among all of these applications, it is essential to determine the location of every sensor with desired accuracy in order to fully realize the functionalities of sensor networks. Although global positioning systems or manual configuration can localize these sensor nodes, they usually require high cost and/or are infeasible to deploy in some scenarios. Therefore, estimating locations of the sensor nodes based on measurements of neighboring nodes has attracted much research interest in recent years [11]–[13]. The measurements could be relative distances or angles, which are usually measured using signal transmitters based on criteria, such as time of arrival, time-difference of arrival, or strength of received radio-frequency signals [14]. Due to limited transmission power, the measurements can only be obtained within a specified range, called radio range. Furthermore, it is assumed that the global position of some nodes is known, referred to as anchors. Then, a sensor network localization (SNL) problem is defined as given the positions of anchors and the measurable information among different sensors, and how to find positions of the remaining of sensor nodes.

Mathematically, the original formulation of a range-based SNL is a nonlinear equality constrained feasibility problem. If the positions of all sensor nodes can be uniquely localized, there exists only one feasible solution to this feasibility problem. By employing the least-square method, it can be converted into a nonconvex optimization problem, which is NP-hard [15]. However, this problem could be ill-posed as there may exist more than one set of noncongruent localization of the sensors satisfying given distance measurements. Thus, two questions naturally arise: first, what conditions lead to a uniquely localizable network and second, how to effectively and efficiently localize all undetermined sensor nodes in a network?

The localizability of a sensor network is usually revealed using graph rigidity theory [16]–[18]. So and Ye [19] showed the problem of deciding whether a given network localization instance is uniquely localizable. They further discovered that the problem of determining the node positions of a uniquely localizable instance can be solved efficiently using semidefinite programming (SDP). The works in [20] and [21] proved that a sensor network is uniquely localizable if and only if the grounded graph is generically globally rigid. However, it is difficult to satisfy the generic global rigidity condition, especially for large-scale networks.

Due to the nonconvex nature of the SNL feasibility problem, many approaches have been proposed. Existing SNL approaches can be classified into centralized [22], [23] and distributed algorithms [11], [24]–[26], based on the structure of the computational framework. In the centralized framework, all distance-based measurements are collected in a data fusion center for processing, and all unknown variables are determined together. Relaxation methods, such as SDP [27], [28] and second-order

Manuscript received May 23, 2019; revised May 27, 2019; accepted June 26, 2019. Date of publication July 4, 2019; date of current version June 12, 2020. This work was supported by the NSF grants ECCS-1815930. Recommended by Associate Editor S. Dey. *(Corresponding author: Ran Dai.)*

The authors are with the Mechanical and Aerospace Engineering Department, The Ohio State University, Columbus, OH 43210 USA (e-mail: wan.326@osu.edu; jing.174@osu.edu; you.242@osu.edu; dai.490@osu.edu).

Digital Object Identifier 10.1109/TCNS.2019.2926775

cone programming (SOCP) [29], have been applied to SNL by converting/relaxing the original problem into a computationally tractable convex optimization problem. However, since relaxation methods ignore the rank constraints, they usually require the grounded graph to be universally rigid (shown as the unique localizability condition in [30]), which is more strict than global rigidity. It is obvious that the centralized framework not only requires a heavy computational load but also consumes a great amount of communication power.

In contrast, distributed algorithms localize all sensor nodes with local information exchanged among neighboring nodes, which is scalable and energy efficient. Hence, distributed algorithms for SNL have attracted more attention, especially for large-scale sensor networks. For example, cluster-based SDP [28], node-based SDP [31], and edge-based SDP [32] are three different distributed formulations based on the SDP method. Srirangarajan *et al.* [11] and Shi *et al.* [33] proposed an edge-based SOCP method. However, each subproblem in these distributed frameworks relies on existing SNL algorithms based on relaxation approaches, which also requires the grounded graph to be universally rigid. In addition, Barycentric coordinate-based distributed algorithms [34], [35] and distributed multidimensional scaling algorithms [36] have been developed to solve the SNL problem with additional assumptions. For example, [34] assumed that any three of each node's neighbors are not collinear. Reference [35] required that the sensors should be located in the convex hull of their neighbors. With good initialization, a gradient-based distributed method [37] claimed to be efficient for solving SNL.

This paper presents some novel results on both the condition for unique localizability and the position-seeking algorithms. On one hand, by considering relationships between nonadjacent sensors, a milder graph condition for unique localizability is derived. We show that a network can be uniquely localized even if the sensing graph is not globally rigid. More precisely, infinitesimal rigidity, together with some mild conditions, is sufficient for unique localizability. To the best of our knowledge, this condition is milder than any works in the range-based SNL literature. On the other hand, an iterative algorithm, called the alternating rank minimization algorithm (ARMA), is proposed to solve rank-constrained SDP programs, which can be applied to solve SNL problems. Compared to the literature of SNL based on SDP methods [22], [23], we handle the rank constraint by equivalently converting it into complementary constraints, which can be solved iteratively with local convergence. As a result, our algorithm seeks an exact solution rather than an approximated one.

Moreover, a distributed algorithm based on ARMA is proposed. The original SNL problem is decomposed into a group of node-based subproblems, where each subproblem has only one node to be localized and all other sensors are fixed using the results obtained from the previous iteration. As every formulated subproblem is a semidefinite problem, it can be solved via ARMA at each iteration. However, since only local information is available for every node when implementing the distributed algorithm, the nonadjacency inequality constraints will not be considered. Furthermore, due to the distributed feature of the algorithm, the dimension of each subproblem is a linear function with respect to the number of its immediate neighbors. Thus, the computational cost per node will not increase in large-scale networks. In addition, simulations in this paper show that the existence of unlocalizable sensors will not affect the estimation accuracy of other localizable sensors.

Throughout this paper, $\mathbb{R}$ denotes the set of real numbers; $\mathbb{S}$ denotes the set of symmetric matrices; $\mathbb{N}$ is the set of positive integers; $\text{col}(A)$, $\text{null}(A)$, $\text{rank}(A)$ denote the column space, null space, and rank of matrix A, respectively; $\dim(M)$ is the dimension of a space M; $|C|$ is the cardinality of set C; $\text{tr}(X)$ is the trace of matrix X; and $\langle A, B\rangle$ denotes the inner product of matrices A and B, that is, $\text{tr}(A^T B)$. $\nabla f(x)$ and $\nabla^2 f(x)$ are the gradient and Hessian matrix of $f(\cdot)$ with respect to x, respectively.

This paper is organized as follows. Section II describes the general formulation of SNL, the graph rigidity conditions for unique localization, and the semidefinite programming relaxation (SDPR) method for SNL. In Section III, the ARMA is proposed and applied to solving SNL in a centralized framework with proof of convergence. Section IV presents the distributed algorithm with computational analysis. In Section V, the numerical simulation results from the proposed methods and comparison with SDP relaxation are presented. The conclusions are addressed in Section VI.

II. Wireless-Sensor Networks Localization

A. Problem Statement

In this paper, we focus on solving the range-based SNL problem. Consider a static sensor network in $\mathbb{R}^s$ $(s = 2 \text{ or } 3)$ has m sensors whose positions are unknown and n anchors whose positions are known $(s < n, m > 0)$. The network of sensors can be indexed by $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_s$, where $\mathcal{V}_a = \{1, 2, \ldots, n\}$ and $\mathcal{V}_s = \{n+1, n+2, \ldots, n+m\}$. In the network, each sensor has the capability of sensing range measurements from other sensors within a fixed range R_0. Let $\mathbf{x}_i \in \mathbb{R}^s$, $i \in \mathcal{V}_s$ be the position of the ith sensor, and $d_{i,j}$ be the Euclidean distance between a pair of sensors $\mathbf{x}_i$ and $\mathbf{x}_j$. Then, the sensing edges can be summarized in a set $\mathcal{E} = \{(i,j) \in \mathcal{V}^2 : \|\mathbf{x}_i - \mathbf{x}_j\| \le R_0\}$. Accordingly, a disk graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$[1] can be used to interpret the sensing relationships between sensors. Let $\mathbf{x} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_{n+m}^T)^T$, $\mathcal{E}_{\text{ss}}$ be the set of edges between nonanchor nodes, and $\mathcal{E}_{\text{as}}$ be the set of edges between anchor nodes and nonanchor nodes.

A range-based SNL problem is to determine the position of sensors $\mathbf{x}_i$, $i \in \mathcal{V}_s$, when all anchors' position $\mathbf{x}_j$, $j \in \mathcal{V}_a$, and available measurements $d_{j,i}$, $(i,j) \in \mathcal{E}$ are given. Mathematically, the SNL problem can be formulated as

$$\begin{aligned} &\text{find}\ \mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^s \\ &\text{such that}\ \forall (i,j) \in \mathcal{E}_{\text{ss}},\ \forall (k,i) \in \mathcal{E}_{\text{as}}. \end{aligned} \tag{1}$$

The above SNL considers noise-free measurements and all anchors' positions are accurate. With the assumption of an ideal scenario, we focus on finding new conditions for unique localizability and developing optimization algorithms to search the

[1]A disk graph is a graph whose edge set is induced by internode distances.

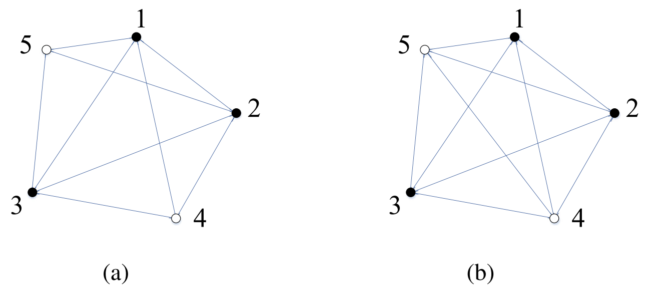


Fig. 1. Illustrative examples of graph rigidity. (a) Globally rigid graph. (b) Universally rigid graph.

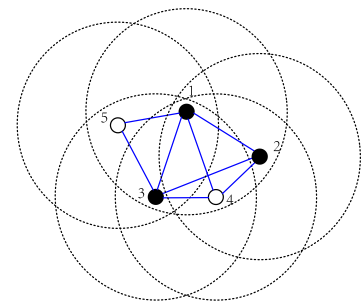


Fig. 2. Graph that can be uniquely localized, but is not globally rigid.

exact solution of problem (1), which can be extended to cases with noisy measurements.

B. Graph Rigidity and Localizability of SNL

Graph rigidity is a useful tool to determine whether the coordinates of nodes in a graph can be uniquely determined (here, two sets of coordinates are considered to be identical if they can be transformed to each other via rigid transformations, that is, translations and rotations) when partial internode distances are known [38], [39]. We use a pair $(\mathcal{G}, \mathbf{x})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathbf{x} = (\mathbf{x}_1^T, \dots, \mathbf{x}_{|\mathcal{V}|}^T)^T \in \mathbb{R}^{|\mathcal{V}|s}$, to describe a framework embedded in $\mathbb{R}^s$. A framework $(\mathcal{G}, \mathbf{x})$ is globally rigid in $\mathbb{R}^s$ if $\mathbf{x} \in \mathbb{R}^{|\mathcal{V}|s}$ can be uniquely determined when distances $||\mathbf{x}_i - \mathbf{x}_j||$, $(i,j) \in \mathcal{E}$ are given. A framework $(\mathcal{G}, \mathbf{x})$ is universally rigid if it is globally rigid for any $s \in \mathbb{N}$. For example, the framework in Fig. 1(a) is globally rigid, while the one in Fig. 1(b) is universally rigid because the shape will not change even when the graph is embedded into three or higher dimensional space.

Another important concept in graph rigidity theory is infinitesimal rigidity, which is also the condition we will focus on for unique localizability. To explain this concept, we first introduce the term of infinitesimal motion. An infinitesimal motion of a framework $(\mathcal{G}, \mathbf{x})$ is a velocity vector $v = \frac{d\mathbf{x}}{dt}$ such that $\frac{d\|\mathbf{x}_i - \mathbf{x}_j\|}{dt} = 0$ for all $(i,j) \in \mathcal{E}$. Then, the definition of *infinitesimal rigidity* is stated below.

Definition 1. (Infinitesimal Rigidity): A framework is infinitesimally rigid if all of the infinitesimal motions are trivial, that is, either rotations or translations.

The SNL problem defined in (1) is uniquely localizable if there is a unique set of sensor locations satisfying the given measurements. In the literature, the localizability of the SNL problem is usually connected with graph rigidity theory. To facilitate the analysis of unique localizability using graph rigidity, the concept of the grounded graph is introduced where an edge exists between any two anchors as their positions are already determined. The overall edge set of the grounded graph is then denoted by $\bar{\mathcal{E}} = \mathcal{E} \cup \{(i,j) \in \mathcal{V}_a^2\}$. Therefore, it is more reasonable to consider the realization of framework $(\mathcal{H}, \mathbf{x})$ with $\mathcal{H} = (\mathcal{V}, \bar{\mathcal{E}})$ as an equivalent problem to the original SNL.

In the existing literature of SNL, for example, [19], by a relaxation approach, the universal rigidity of $(\mathcal{H}, \mathbf{x})$ is considered as the necessity and sufficiency condition for unique localizability. In [20] and [21], without ignoring the rank constraint, a milder condition, that is, global rigidity of $(\mathcal{H}, \mathbf{x})$, has been recognized as the necessity and sufficiency condition for unique localizability. In these works, an important property of disk graphs has always been ignored. That is, the distance between each pair of nonadjacent nodes should be greater than R_0. More precisely, in existing centralized methods, the constraints on nonadjacent nodes, $||\mathbf{x}_i - \mathbf{x}_j|| \geq R_0$, $(i,j) \in \mathcal{V}^2 \setminus \bar{\mathcal{E}}$, have not been considered.

In fact, these constraints are quite useful in sensor localization. When considering additional constraints on nonadjacent nodes, the framework $(\mathcal{H}, \mathbf{x})$ is not necessary to be globally rigid for unique localization. For example, in Fig. 2, nodes 1, 2, and 3 are anchors, and 4 and 5 are sensors whose positions are to be determined. It is obvious that the grounded graph is not globally rigid since node 5 can be placed on the other side of edge $(1,3)$ while keeping all internode distances invariant. However, when node 5 is in the incorrect position, it would become the neighbor of node 4, which violates the constraint $||\mathbf{x}_4 - \mathbf{x}_5|| \geq R_0$. That is, the sensor network in Fig. 2 can be uniquely localized under constraints from measured distances and nonadjacency constraints. Since these inequalities from nonadjacency relationships are shared globally among all network nodes, they will be considered in the proposed centralized algorithm.

C. SDP Formulation of SNL

Solving the feasibility problem in (1) directly is difficult due to the nonconvex equalities from measured distance. One approach is to introduce unknown matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m] \in \mathbb{R}^{s\times m}$ and $\mathbf{Y} = \mathbf{X}^T\mathbf{X}$ to rewrite the original SNL problem in (1) as a feasibility problem of finding unknown matrices, expressed as follows:

$$
\begin{aligned}
&\text{find } \mathbf{X} \in \mathbb{R}^{s\times m}, \ \mathbf{Y} \in \mathbb{R}^{m\times m} \\
&\text{such that } (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{Y} (\mathbf{e}_i - \mathbf{e}_j) = d_{i,j}^2, \forall (i,j) \in \mathcal{E}_{\text{ss}} \\
&\qquad \begin{bmatrix} \mathbf{x}_k \\ -\mathbf{e}_i \end{bmatrix}^T \begin{bmatrix} \mathbf{I}_s & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Y} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ -\mathbf{e}_i \end{bmatrix} = d_{k,i}^2, \forall (k,i) \in \mathcal{E}_{\text{as}} \\
&\qquad \mathbf{Y} = \mathbf{X}^T\mathbf{X},
\end{aligned} \tag{2}
$$

where $\mathbf{e}_i \in \mathbb{R}^m$ is a unit vector with zero entries except the ith entry and $\mathbf{I}_s \in \mathbb{R}^{s\times s}$ is an identity matrix. By introducing

$$
\mathbf{Z} = \begin{bmatrix} \mathbf{I}_s & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Y} \end{bmatrix} \in \mathbb{R}^{(s+m)\times(s+m)} \tag{3}
$$

the feasibility problem in (2) can be equivalently reformulated as an optimization problem with a rank constraint on $\mathbf{Z}$

$$\begin{aligned}
&\min_{\mathbf{Z}} \ 0 \\
&\quad \text{s.t. } \mathbf{Z}_{(1:s,1:s)} = \mathbf{I}_s \\
&\quad [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z} [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j] = d_{i,j}^2, \forall (i,j) \in \mathcal{E}_{\text{ss}} \\
&\quad [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z} [\mathbf{x}_k; -\mathbf{e}_i] = d_{k,i}^2, \forall (k,i) \in \mathcal{E}_{\text{as}} \\
&\quad \mathbf{Z} \succeq 0, \ \text{rank}(\mathbf{Z}) = s
\end{aligned} \tag{4}$$

where $\mathbf{Z}_{(1:s,1:s)}$ represents the upper-left s-dimensional principle submatrix of $\mathbf{Z}$, and $\mathbf{Z} \succeq 0$ refers to a positive semidefinite matrix. Since $\mathbf{Z}_{(1:s,1:s)} = \mathbf{I}_s$, it implies that $\text{rank}(\mathbf{Z}) \geq s$. Thus, the rank equality constraint $\text{rank}(\mathbf{Z}) = s$ is equivalent to $\text{rank}(\mathbf{Z}) \leq s$.

In the semidefinite relaxation approach, the rank constraint $\text{rank}(\mathbf{Z}) = s$ in (4) is ignored. Then, problem (4) can be relaxed as the following convex SDP problem:

$$\begin{aligned}
&\min_{\mathbf{Z}} \ 0 \\
&\quad \text{s.t } \mathbf{Z}_{(1:s,1:s)} = \mathbf{I}_s \\
&\quad [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z} [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j] = d_{i,j}^2, \forall (i,j) \in \mathcal{E}_{\text{ss}} \\
&\quad [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z} [\mathbf{x}_k; -\mathbf{e}_i] = d_{k,i}^2, \forall (k,i) \in \mathcal{E}_{\text{as}} \\
&\quad \mathbf{Z} \succeq 0.
\end{aligned} \tag{5}$$

This relaxed problem can be solved via an SDP solver, such as Sedumi [40]. Besides, So and Ye [19] showed that the relaxed SDP leads to a unique solution if and only if the grounded graph is universally rigid.

III. Iterative Framework for Centralized SNL

A. Alternating Rank Minimization Algorithm

Consider the following general rank-constrained SDP problem:

$$\begin{aligned}
&\min_{\mathbf{X}} \ \langle \mathbf{Q}_0, \mathbf{X} \rangle \\
&\quad \text{s.t. } \langle \mathbf{Q}_i, \mathbf{X} \rangle = c_i, i = 1, 2, \ldots, N \\
&\quad \langle \mathbf{Q}_i, \mathbf{X} \rangle \leq c_i, i = N+1, N+2, \ldots, \hat{N} \\
&\quad \text{rank}(\mathbf{X}) \leq r, \ \mathbf{X} \in \mathbb{S}_+^n
\end{aligned} \tag{6}$$

where $\mathbf{X} \in \mathbb{S}_+^n$ is a general unknown positive semidefinite matrix and $\mathbf{Q}_i \in \mathbb{R}^{n\times n}$, $i = 0, 1, \ldots, \hat{N}$ are real symmetric coefficient matrices, which are not necessarily positive definite, $c_i \in \mathbb{R}$ is a constant item in the linear matrix constraint, and r is a given positive integer. As we explained that the rank equality constraint in (4) is equivalent to a rank inequality constraint, the SNL problem in (4) in terms of $\mathbf{Z}$ can be generalized in the form of (6).

Instead of ignoring the rank constraint in (6), we replace the rank constraint $\text{rank}(\mathbf{X}) \leq r$ by an alternative set of constraints, expressed as follows:

$$\begin{aligned}
&\langle \mathbf{W}, \mathbf{I}_n \rangle = n - r, \ \mathbf{W} \in \mathbb{S}_+^n \\
&\mathbf{X} \perp \mathbf{W}, \ \mathbf{X} \in \mathbb{S}_+^n \\
&\mathbf{I}_n - \mathbf{W} \in \mathbb{S}_+^n
\end{aligned} \tag{7}$$

where $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix. $\mathbf{X} \perp \mathbf{W}$ indicates that $\mathbf{W}$ is the orthogonal complement of $\mathbf{X}$, that is, $\langle \mathbf{X}, \mathbf{W} \rangle = 0$. The idea of reformulating constraints into complementary conditions, as shown in (7), is commonly used in the literature of optimization, for example, [41]. We give the following theorem to show the equivalence between the rank constraint and constraints stated in (7).

Theorem 3.1: The rank constraint $\text{rank}(\mathbf{X}) \leq r$ for $\mathbf{X} \in \mathbb{S}_+^n$ is equivalent to the set of constraints stated in (7).

Proof: Letting $\mathbf{U} = I - \mathbf{W}$, it is easy to see that the constraints in (7) are equivalent to

$$\begin{aligned}
&\langle \mathbf{X}, \mathbf{U} \rangle = \text{trace}(\mathbf{X}) \\
&\langle \mathbf{I}, \mathbf{U} \rangle = r \\
&\mathbf{0} \preceq \mathbf{U} \preceq \mathbf{I}.
\end{aligned}$$

The equivalence between the rank constraint and the abovementioned equations has been given in [41, Theor. 1.1]. ■

Considering the equivalent relationship between the rank constraint and the set of constraints in (7), problem (6) can be equivalently reformulated as follows:

$$\begin{aligned}
&\min_{\mathbf{X},\mathbf{W}} \ \langle \mathbf{Q}_0, \mathbf{X} \rangle \\
&\quad \text{s.t. } \langle \mathbf{Q}_i, \mathbf{X} \rangle = c_i, \ i = 1, 2, \ldots, N \\
&\quad \langle \mathbf{Q}_i, \mathbf{X} \rangle \leq c_i, i = N+1, N+2, \ldots, \hat{N} \\
&\quad \langle \mathbf{W}, \mathbf{I} \rangle = n - r, \ \mathbf{W} \in \mathbb{S}_+^n \\
&\quad \langle \mathbf{X}, \mathbf{W} \rangle = 0, \ \mathbf{X} \in \mathbb{S}_+^n \\
&\quad \mathbf{I} - \mathbf{W} \in \mathbb{S}_+^n.
\end{aligned} \tag{8}$$

Although the rank constraint is now replaced by its equivalence, (8) is still a nonconvex problem due to the bilinear constraint $\langle \mathbf{X}, \mathbf{W} \rangle = 0$. Thus, we propose an alternating method to solve $\mathbf{X}$ and $\mathbf{W}$ separately by decomposing it into two convex subproblems. Moreover, the matrix equality constraint $\langle \mathbf{X}, \mathbf{W} \rangle = 0$ will be relaxed as penalty terms in the objective function. Specifically, assuming at the kth step, $\mathbf{W} = \mathbf{W}_{k-1}$ is given, problem (8) becomes a convex problem with respect to $\mathbf{X}$, formulated as

$$\begin{aligned}
&\min_{\mathbf{X}} \ \langle \mathbf{Q}_0, \mathbf{X} \rangle + \alpha_k \langle \mathbf{X}, \mathbf{W}_{k-1} \rangle \\
&\quad \text{s.t. } \langle \mathbf{Q}_i, \mathbf{X} \rangle = c_i, i = 1, 2, \ldots, N \\
&\quad \langle \mathbf{Q}_i, \mathbf{X} \rangle \leq c_i, i = N+1, N+2, \ldots, \hat{N} \\
&\quad \mathbf{X} \succeq 0.
\end{aligned} \tag{9}$$

where α_k is a positive weighting factor. Let $\mathbf{X}_k$ be the optimal solution of (9), i.e.,

$$\mathbf{X}_k = \arg \min_{\mathbf{X} \in \mathcal{C}_X^k} \langle \mathbf{Q}_0, \mathbf{X} \rangle + \alpha_k \langle \mathbf{X}, \mathbf{W}_{k-1} \rangle \tag{10}$$

where $\mathcal{C}_X^k$ is the set of constraints in (9).

Algorithm 1: Alternating Rank Minimization Algorithm.

```
Input: Q_i, c_i, i = 0, 1, ..., N̂, ϵ, k_max
Output: Local optimum X.
begin
    Initialize W_0 with X_0 from SDP relaxation of (6).
    for k ← 1 to k_max do
        update X_k by solving (9);
        update W_k by solving (11);
        if ⟨X_{k+1}, W_{k+1}⟩ ≤ ϵ then
            Stop and output X;
        else
            Update k ← k + 1
        end if
    end for
end
```

With a given $\mathbf{X} = \mathbf{X}_k$, problem (8) becomes a convex problem with respect to $\mathbf{W}$. Again, considering the equality constraint $\langle \mathbf{X}, \mathbf{W} \rangle = 0$ as penalty terms, we have

$$\begin{aligned} \min_{\mathbf{W}} \ & \langle \mathbf{Q}_0, \mathbf{X}_k \rangle + \alpha_k \langle \mathbf{X}_k, \mathbf{W} \rangle \\ \text{s.t.} \ & \langle \mathbf{W}, \mathbf{I} \rangle = n - r \\ & \mathbf{I} - \mathbf{W} \succeq 0, \ \mathbf{W} \succeq 0. \end{aligned} \tag{11}$$

Similarly, $\mathbf{W}_k$ is denoted by

$$\mathbf{W}_k = \arg \min_{\mathbf{W} \in \mathcal{C}_W^k} \langle \mathbf{Q}_0, \mathbf{X}_k \rangle + \alpha_k \langle \mathbf{X}_k, \mathbf{W} \rangle \tag{12}$$

where $\mathcal{C}_W^k$ is the set of constraints in (11).

By giving an initial value $\mathbf{X}_0$, (9) and (11) can be solved iteratively until it satisfies the stopping criterion. The ARMA is summarized in Algorithm 1.

B. Convergence Analysis of ARMA

This section presents the convergence analysis of the proposed ARMA. Denote $\omega_k = \langle \mathbf{X}_k, \mathbf{W}_k \rangle$ and $J_k = \langle \mathbf{Q}_0, \mathbf{X}_k \rangle + \alpha_k \omega_k$. In addition, the compact forms of (10) and (12) can be represented by two mappings $\Gamma_1(\mathbf{W}) : \mathbb{S}_n^+ \mapsto \mathbb{S}_n^+$ and $\Gamma_2(\mathbf{X}) : \mathbb{S}_n^+ \mapsto \mathbb{S}_n^+$, respectively. Thus, we have

$$\mathbf{X}_k = \Gamma_1(\mathbf{W}_{k-1}), \ \mathbf{W}_k = \Gamma_2(\mathbf{X}_k). \tag{13}$$

Assumption 3.2: Problem (6) is feasible.

Assumption 3.3: $\hat{\mathbf{Q}} := \sum_1^N \mathbf{Q}_i \mathbf{Q}_i^T$ is positive definite, that is, $\text{rank}(\hat{\mathbf{Q}}) = n$.

We first define

$$\mathcal{L}(\mathbf{X}, \mathbf{W}) = \langle \mathbf{Q}_0, \mathbf{X} \rangle + \alpha \langle \mathbf{X}, \mathbf{W} \rangle + \beta (n - r - \langle \mathbf{I}, \mathbf{W} \rangle)^2 + \beta \sum_{i=1}^N (\langle \mathbf{Q}_i, \mathbf{X} \rangle - c_i)^2$$

where $\alpha > 0$ is a weighting factor, $\beta > \frac{\alpha}{2\sqrt{\lambda_{\min}}}$, and $\lambda_{\min} > 0$ is the smallest eigenvalue of $\hat{\mathbf{Q}}$. In addition, $n - r - \langle \mathbf{I}, \mathbf{W} \rangle = 0$ for any $\mathbf{W} = \mathbf{W}_k$ obtained from (12), and $\langle \mathbf{Q}_i, \mathbf{X} \rangle - c_i = 0$ for any $\mathbf{X} = \mathbf{X}_k$ obtained from (10). Next, we denote

$$\mathbf{A} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{X}^2} = 2\beta \sum_{i=1}^N \mathbf{Q}_i \mathbf{Q}_i^T = 2\beta \hat{\mathbf{Q}}$$

$$\mathbf{B} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{W}\mathbf{X}} = \alpha \mathbf{I}$$

$$\mathbf{C} = \frac{\partial^2 \mathcal{L}}{\partial \mathbf{W}^2} = 2\beta \mathbf{I}.$$

Lemma 3.4: With Assumption 3.3 holding, $\Gamma_1(\mathbf{W})$ and $\Gamma_2(\mathbf{X})$ are continuously differentiable in some neighborhood of $(\mathbf{X}^*, \mathbf{W}^*)$.

Proof: Let $(\mathbf{X}^*, \mathbf{W}^*)$ be a pair of local optimal solutions of (8), since $\mathcal{L}(\mathbf{X}, \mathbf{W})$ is twice differentiable, and $\Gamma_1(\mathbf{W})$ and $\Gamma_2(\mathbf{X})$ are continuously differentiable in some neighborhood of $(\mathbf{X}^*, \mathbf{W}^*)$. ■

Remark 1: When Assumption 3.3 holds, both $\mathbf{A}$ and $\mathbf{C}$ are positive definite. In addition, for the local optimal solution $(\mathbf{X}^*, \mathbf{W}^*)$, $\langle \mathbf{X}^*, \mathbf{W}^* \rangle = 0$ holds, then $\mathbf{X}^* \in \Gamma_1(\mathbf{W}^*), \mathbf{W}^* \in \Gamma_2(\mathbf{X}^*)$, that is, $\mathbf{X}^*$ will be the optimum of the subproblem (10).

Define two maps $\Upsilon_1 : \mathbb{S}_n^+ \mapsto \mathbb{S}_n^+$ and $\Upsilon_2 : \mathbb{S}_n^+ \mapsto \mathbb{S}_n^+$ as

$$\mathbf{X}_k = \Upsilon_1(\mathbf{X}_{k-1}) = \Gamma_1(\mathbf{W}_{k-1}) = \Gamma_1(\Gamma_2(\mathbf{X}_{k-1}))$$

$$\mathbf{W}_k = \Upsilon_2(\mathbf{W}_{k-1}) = \Gamma_2(\mathbf{X}_k) = \Gamma_2(\Gamma_1(\mathbf{W}_{k-1})).$$

Lemma 3.5: With Assumption 3.3 holding and $\mathcal{L}(\mathbf{X}, \mathbf{W})$ being twice differentiable, $\rho(\Upsilon_1'(\mathbf{X}^*)) = \rho(\Upsilon_2'(\mathbf{W}^*)) < 1$, where $\rho(\bullet)$ is the spectral radius of matrix "$\bullet$."

Proof: First, with $\beta > \frac{\alpha}{2\sqrt{\lambda_{\min}}}$, the Hessian matrix of $\mathcal{L}$ at $(\mathbf{X}^*, \mathbf{W}^*)$ will be positive definite, i.e.,

$$\nabla^2 \mathcal{L} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} = \begin{bmatrix} 2\beta\hat{\mathbf{Q}} & \alpha \mathbf{I} \\ \alpha \mathbf{I} & 2\beta \mathbf{I} \end{bmatrix} \succ \mathbf{0}. \tag{14}$$

Based on the Schur complement of a block matrix, (14) is equivalent to

$$\begin{aligned} & 2\beta\hat{\mathbf{Q}} \succ \mathbf{0}, \ 2\beta \mathbf{I} \succ \mathbf{0} \\ & 2\beta \mathbf{I} - \alpha \mathbf{I} (2\beta \hat{\mathbf{Q}})^{-1} \alpha \mathbf{I} = 2\beta \mathbf{I} - \frac{\alpha^2}{2\beta} \hat{\mathbf{Q}}^{-1} \succ \mathbf{0} \\ \Leftrightarrow \ & 2\beta \left(\mathbf{I} - \frac{\alpha^2}{4\beta^2} \hat{\mathbf{Q}}^{-1} \right) \succ \mathbf{0} \\ \Leftrightarrow \ & \rho \left(\frac{\alpha^2}{4\beta^2} \hat{\mathbf{Q}}^{-1} \right) < 1. \end{aligned} \tag{15}$$

By differentiating (13), we get $\Upsilon_1'(\mathbf{X}) = \Gamma_1' \Gamma_2'(\mathbf{X})$ and then differentiating $\frac{\partial \mathcal{L}(\mathbf{X}, \Gamma_2(\mathbf{X}))}{\partial (\mathbf{W})}$ with respect to $\mathbf{X}$ and evaluating at $\mathbf{X}^*$ leads to

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{W}^2} \Gamma_2'(\mathbf{X}^*) + \alpha \mathbf{I} = 2\beta \mathbf{I} \Gamma_2'(\mathbf{X}^*) + \alpha \mathbf{I} = 0$$

$$\Leftrightarrow \Gamma_2'(\mathbf{X}^*) = -\frac{\alpha}{2\beta} \mathbf{I}.$$

Similarly, it is easy to obtain

$$\frac{\partial^2 \mathcal{L}}{\partial \mathbf{X}^2}\Gamma_1'(\mathbf{W}^*) + \alpha\mathbf{I} = 2\beta\hat{\mathbf{Q}}\Gamma_1'(\mathbf{W}^*) + \alpha\mathbf{I} = 0$$
$$\Leftrightarrow \Gamma_1'(\mathbf{X}^*) = -\frac{\alpha}{2\beta}\hat{\mathbf{Q}}^{-1}.$$

Therefore

$$\Upsilon_1'(\mathbf{X}^*) = \Upsilon_2'(\mathbf{W}^*) = \frac{\alpha^2}{4\beta^2}\hat{\mathbf{Q}}^{-1}. \tag{16}$$

Combining (15) completes the proof. ∎

Theorem 3.6: (Local Convergence) If α_k is nondecreasing and upper bounded, let $(\mathbf{X}^*, \mathbf{W}^*)$ be a local minimum of (8), then for any starting point $(\mathbf{X}_k, \mathbf{W}_k)$ in some neighborhood of $(\mathbf{X}^*, \mathbf{W}^*)$, the sequence $\{(\mathbf{X}_k, \mathbf{W}_k)\}$ generated by Algorithm 1 will converge Q-linearly to $(\mathbf{X}^*, \mathbf{W}^*)$.

Proof: In Lemma 3.5, we shows that the Hessian matrix of the objective function $\mathcal{L}$ is positive definite, and $\rho = \rho(\Upsilon_1'(\mathbf{X}^*)) = \rho(\Upsilon_2'(\mathbf{W}^*)) < 1$. For $\rho < 1$, there exists $\delta > 0$ such that $\rho + 2\delta < 1$. Based on the definition of spectral radius of a matrix and continuity of $\Upsilon_1'(\mathbf{X})$ at the neighborhood of $\mathbf{X}^*$, there exists a matrix norm $\|\bullet\|_\delta$ (depending on δ) such that

$$\|\Upsilon_1'(\mathbf{X})(\mathbf{X} - \mathbf{X}^*)\| \le (\rho + \delta)\|\mathbf{X} - \mathbf{X}^*\|. \tag{17}$$

Then, starting with $\mathbf{X}^* = \Upsilon_1(\mathbf{X}^*)$, we have

$$\begin{aligned}
\|\mathbf{X}_{k+1} - \mathbf{X}^*\|_\delta &= \|\Upsilon_1(\mathbf{X}_k) - \Upsilon_1(\mathbf{X}^*)\|_\delta \\
&= \|\int_{\mathbf{X}^*}^{\mathbf{X}_k} \Upsilon_1'(\mathbf{X})\mathrm{d}\mathbf{X}\|_\delta \\
&= \|\int_0^1 \Upsilon_1'(t(\mathbf{X} - \mathbf{X}^*) + \mathbf{X}^*)(\mathbf{X}_k - \mathbf{X}^*)\mathrm{d}t\|_\delta \\
&\le \int_0^1 \|\Upsilon_1'(t(\mathbf{X} - \mathbf{X}^*) + \mathbf{X}^*)(\mathbf{X}_k - \mathbf{X}^*)\|_\delta \mathrm{d}t \\
&\le (\rho + 2\delta)\|\mathbf{X}_k - \mathbf{X}^*\|_\delta
\end{aligned} \tag{18}$$

where the third equality holds by using the transformation $\mathbf{X} = t(\mathbf{X}_k - \mathbf{X}^*) + \mathbf{X}^*$ with $t \in [0, 1]$. The last inequality holds based the fact that when $t \in [0, 1]$ and $\mathbf{X}_k$ is in the neighborhood of $\mathbf{X}^*$, then $\mathbf{X} = (t(\mathbf{X}_k - \mathbf{X}^*) + \mathbf{X}^*)$ will also be in the neighborhood of $\mathbf{X}^*$, thus the inequality (17) holds for $\mathbf{X}$. A similar result for $\mathbf{W}_k$ can be proved by using the same scheme. In summary, the sequences $\{\mathbf{X}_k\}, \{\mathbf{W}_k\}$ generated by Algorithm 1 will converge to $\mathbf{X}^*, \mathbf{W}^*$, respectively. ∎

Remark 2: The above theorem shows that each local minimum (including the global minimum) is locally stable. In the simulation experiments, however, algorithm 1 can always drive a random initial guess to the global minimum of the SNL problem. We leave the global convergence proof as one of our future research directions.

C. Centralized SNL Framework Based on ARMA

The proposed ARMA for solving rank-constrained SDP problems is first applied to SNL in a centralized framework. As we described in Section II-B, considering the inequality conditions of nonadjacent nodes, that is, $||\mathbf{x}_i - \mathbf{x}_j|| \ge R_0$, $(i, j) \in \mathcal{V}^2 \setminus \bar{\mathcal{E}}$ will contribute to additional information for sensor localization. With these additional nonadjacency constraints, problem (4) can be reformulated as follows:

$$\begin{aligned}
&\min_{\mathbf{Z}} \; 0 \\
&\text{s.t. } \mathbf{Z}_{(1:s,1:s)} = \mathbf{I}_s, \; \mathbf{Z} \in \mathbb{S}_+^n \\
&\quad [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z}[\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j] = d_{i,j}^2, \forall (i, j) \in \mathcal{E}_{\text{ss}} \\
&\quad [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z}[\mathbf{x}_k; -\mathbf{e}_i] = d_{k,i}^2, \forall (k, i) \in \mathcal{E}_{\text{as}} \\
&\quad [\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z}[\mathbf{0}_{s\times 1}; \mathbf{e}_i - \mathbf{e}_j] \ge R_0^2, \forall (i, j) \in (\bar{\mathcal{E}} \setminus \mathcal{E}_{\text{ss}}) \\
&\quad [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z}[\mathbf{x}_k; -\mathbf{e}_i] \ge R_0^2, \forall (k, i) \in (\bar{\mathcal{E}} \setminus \mathcal{E}_{\text{as}}) \\
&\quad \text{rank}(\mathbf{Z}) = s.
\end{aligned} \tag{19}$$

Based on the general formulation of rank-constrained SDP, problem (19) can be rewritten as follows:

$$\begin{aligned}
&\min_{\mathbf{Z}} \; 0 \\
&\text{s.t. } \mathbf{Z}_{(1:s,1:s)} = \mathbf{I}_s \\
&\quad \langle \mathbf{Q}_i, \mathbf{Z}\rangle = c_i, i = 1, 2, \ldots, N \\
&\quad \langle \mathbf{Q}_j, \mathbf{Z}\rangle \ge R_0^2, j = 1, 2, \ldots, b \\
&\quad \text{rank}(\mathbf{Z}) = s, \; \mathbf{Z} \in \mathbb{S}_+^n
\end{aligned} \tag{20}$$

where b is the number of nonadjacency constraints. That is, $b = \frac{(n+m)(n+m-1)}{2} - |\bar{\mathcal{E}}|$.

In addition, since the graph of an infinitesimal rigid network is connected and there is at least one anchor, according to [42, Lemma 3], Assumption 3.3 is satisfied for the SNL problem. Therefore, by applying ARMA described in Algorithm 1, problem (20) is solved alternatively until it converges. Furthermore, by considering the inequality constraints on nonadjacent nodes of a network, new sufficient conditions for unique localizability can be achieved. To state the new conditions, the following lemma is first given.

Lemma 3.7: If a framework $(\mathcal{G}, \mathbf{x})$ is infinitesimally rigid in $\mathbb{R}^s$, then for each $i \in \mathcal{V}$, all neighbors of i cannot lie in a hyperplane of $\mathbb{R}^{s-1}$.

Proof: Assuming that all neighbors of i lie in a hyperplane of $\mathbb{R}^{s-1}$, we prove the statement by constructing a nontrival infinitesimal motion for framework $(\mathcal{G}, \mathbf{x})$. Without loss of generality, letting $\mathbf{x}_1, \ldots, \mathbf{x}_l$ be the neighbors of $\mathbf{x}_i$, it is easy to show that $\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_i$ must lie in a hyperplane of $\mathbb{R}^s$. We refer to this hyperplane as M. Let $\eta \in \mathbb{R}^s$ be a normal vector of M, then $a = (0, \ldots, 0, \eta^T, 0\ldots 0)^T \in \mathbb{R}^{ns}$ should be an infinitesimal motion of the framework $(\mathcal{G}, \mathbf{x})$, where η^T includes the $(i-1)s + 1$th to $i s$th components of a. It is obvious that a is neither a rotation nor a translation, which implies that a is nontrivial. A contradiction with infinitesimal rigidity of $(\mathcal{G}, \mathbf{x})$ arises. ∎

Let $H_{y_k} = I_s - 2y_k y_k^T$ be the Householder transformation, y_k is a unit vector that is orthogonal to the hyperplane determined by $\mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_l}$, $k_i \in \mathcal{N}_k$, $l = |\mathcal{N}_k|$. With Lemma 3.7 stated above, we can give the following result.

Theorem 3.8: Suppose that $(\mathcal{H}, \mathbf{x})$ is infinitesimally rigid, the sensor network is uniquely localizable if and only if for any $k \in \mathcal{V}_s$, one of the following two conditions is satisfied.

i) There exist $s+1$ nodes in $\mathcal{N}_k$ not lying in a hyperplane of $\mathbb{R}^s$.

ii) Condition (i) is invalid. There exists a node $i \in \mathcal{V} \setminus \mathcal{N}_k$, such that $||\mathbf{x}_i - H_{y_k}\mathbf{x}_k|| < R_0$.

Proof. Sufficiency: We prove the result by induction. When $|\mathcal{V}| = 3$, since we have stipulated that $n > s \geq 2$, all three sensors are anchors, and it is straightforward that the network can be uniquely localized.

Assume that the network $(\mathcal{H}, \mathbf{x})$ with $|\mathcal{V}| = N$ sensors can be uniquely localized, where $\mathcal{H} = (\mathcal{V}_s \cup \mathcal{V}_a, \mathcal{E})$. Next, we show that after adding a sensor k and related edges satisfying condition (i) or (ii), the induced network $(\bar{\mathcal{H}}, \mathbf{x})$ with $|\mathcal{V}| = N+1$ sensors can still be uniquely localized, where $\bar{\mathcal{H}} = (\bar{\mathcal{V}}_s \cup \mathcal{V}_a, \mathcal{E} \cup \mathcal{E}_k)$, $\mathcal{E}_k$ includes all edges involving k. We consider the following two cases.

Case 1: Condition i) is satisfied. Without loss of generality, let $t_1, \ldots, t_{s+1}$ be the $s+1$ nodes in $\mathcal{N}_k$ not lying in a hyperplane of $\mathbb{R}^s$. Then, we have

$$||\mathbf{x}_k - \mathbf{x}_{t_i}||^2 = d_{ik}^2, \quad i = 1, \ldots, s+1. \tag{21}$$

By a subtraction between the first equation and each equation in (21), for $i = 1, \ldots, s$, we obtain

$$2\sum_{j=1}^{s}(\mathbf{x}_{t_{i+1}j} - \mathbf{x}_{1j})\mathbf{x}_{kj} = d_{t_{i+1}k}^2 - d_{1k}^2 + \sum_{j=1}^{s}(\mathbf{x}_{t_{i+1}j}^2 - \mathbf{x}_{1j}^2) \tag{22}$$

where $\mathbf{x}_{kj}$, is the jth component of $\mathbf{x}_k$, and $\mathbf{x}_{ij}$ is the jth component of $\mathbf{x}_i$.

Let $\mathbf{x}_{k1}, \ldots, \mathbf{x}_{ks}$ be unknown quantities and $A = (\mathbf{x}_{t_2} - \mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_{s+1}} - \mathbf{x}_{t_1})^T \in \mathbb{R}^{s\times s}$, then the linear equality in (22) can be written as $A\mathbf{x}_k = \tilde{d}$, where $\tilde{d} = (\ldots, d_{t_{i+1}k}^2 - d_{1k}^2 + \sum_{j=1}^{s} (\mathbf{x}_{t_{i+1}j}^2 - \mathbf{x}_{1j}^2), \ldots)^T$.

The nondegeneracy of $\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_{s+1}}$ implies that $A \in \mathbb{R}^{s\times s}$ is of full rank. According to Cramer's rule, $\mathbf{x}_k$ can be uniquely determined.

Case 2: Condition (ii) is satisfied. Since condition (i) is invalid, by Lemma 3.7, $\mathbf{x}_{k_1}, \ldots, \mathbf{x}_{k_l}$ determine a unique $(s-1)$-dimensional hyperplane M. It is obvious that $H_{y_k}\mathbf{x}_k$ satisfies all distance constraints. Next, we show that $H_{y_k}\mathbf{x}_k$ is the only possible undesired position of k. Suppose $\mathbf{x}'_k$ is a coordinate distinct to $\mathbf{x}_k$ satisfying all distance constraints, since there are at most two points in a line having the same distance from one point, it suffices to show that $\mathbf{x}_k - \mathbf{x}'_k$ is perpendicular to M.

Note that for any $i \in \{1, \ldots, l\}$, we have $||\mathbf{x}_k - \mathbf{x}_{k_i}||^2 = ||\mathbf{x}'_k - \mathbf{x}_{k_i}||^2$, which implies $(\frac{\mathbf{x}_k + \mathbf{x}'_k}{2} - \mathbf{x}_{k_i})^T(\mathbf{x}_k - \mathbf{x}'_k) = 0$. Hence, for any $i, j \in \{1, \ldots, l\}$, there holds $(\mathbf{x}_{k_i} - \mathbf{x}_{k_j})^T(\mathbf{x}_k - \mathbf{x}'_k) = 0$. We then conclude that $\mathbf{x}_k - \mathbf{x}'_k$ is perpendicular to M. It follows that $\mathbf{x}'_k = H_{y_k}\mathbf{x}_k$. Note also that for any $i \in \mathcal{V} \setminus \mathcal{N}_k$, there must hold $||\mathbf{x}_i - \mathbf{x}_k|| \geq R_0$. Since the condition $||\mathbf{x}_i - H_{y_k}\mathbf{x}_k|| < R_0$ prevents $\mathbf{x}_k$ from being $\mathbf{x}'_k$, $\mathbf{x}_k$ can be uniquely determined.

Necessity: Suppose that conditions i) and ii) do not hold. From the proof for sufficiency, it is straightforward that $\mathbf{x}'_k = H_{y_k}\mathbf{x}_k$ is an undesired coordinate satisfying all distance constraints. That is, the sensor network is not uniquely localizable. ■

One can observe that although the framework in Fig. 2 is not globally rigid, it satisfies the condition in Theorem 3.8.

IV. Distributed Framework for SNL

In this section, a node-based distributed algorithm for SNL is proposed, where the original problem is decomposed into a group of subproblems and each subproblem is solved independently using the proposed ARMA. In the node-based distributed algorithm, it is assumed that each node has access to local information only. In other words, it can only obtain the measurable distances of anchors and sensors located within its radio range. For example, denoting the position vector of the ith sensor at the pth iteration as $\mathbf{x}_i^{(p)} \in \mathbb{R}^s$, then the ith subproblem is formulated as follows:

$$\begin{aligned}
&\min_{\mathbf{x}_i^{(p)}} \ 0 \\
&\text{s.t. } ||\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p-1)}||^2 = d_{i,j}^2, \forall j \in \mathcal{V}_s^i \\
&\qquad ||\mathbf{x}_i^{(p)} - \mathbf{x}_k||^2 = \hat{d}_{k,i}^2 . \forall k \in \mathcal{V}_a^i
\end{aligned} \tag{23}$$

where $\mathbf{x}_j^{(p-1)} \in \mathbb{R}^s$ denotes the estimation of the jth node's position vector at the $(p-1)$th step, $\mathcal{V}_s^i$ refers to the set of adjacent sensors of ith sensor with $(i,j) \in \mathcal{E}_{ss}$, and $\mathcal{V}_a^i$ refers to the set of adjacent anchors of the ith sensor with $(i,k) \in \mathcal{E}_{as}$. Since the nonadjacency constraints belong to global information, they will not be involved in the distributed algorithm. Under this venue, the SNL problem in (1) is decomposed into m subproblems. However, any low precision or incorrect estimation at the $(p-1)$th step may lead to an infeasible solution of (23). Therefore, to guarantee the feasibility of each subproblem, the quadratic equality constraints expressed in (24) are relaxed by introducing a new variable vector $\mathbf{l}_i$, $i = 1, \ldots, m$, such that

$$||\mathbf{x}_i^{(p)} - \mathbf{x}_j^{(p-1)}||^2 = (l_{i,j}^{(p)})^2, \ \forall j \in \mathcal{V}_s^i \tag{24}$$

where $l_{i,j}^{(p)}$ is the jth element of $\mathbf{l}_i$ at the pth iteration. Then, problem (24) can be reformulated as an optimization problem to minimize the differences between $l_{i,j}^{(p)}$ and $d_{i,j}$, expressed as follows:

$$\begin{aligned}
&\min_{\mathbf{x}_i^{(p)}, \mathbf{l}_i^{(p)}} \sum_{j \in \mathcal{V}_s^i} |l_{i,j}^{(p)} - d_{i,j}|^2 \\
&\text{s.t. } ||\mathbf{x}_k - \bar{\mathbf{x}}_i^{(p)}||^2 = \hat{d}_{k,i}^2, k \in \mathcal{V}_a^i \\
&\qquad ||\bar{\mathbf{x}}_i^{(p)} - \bar{\mathbf{x}}_j^{(p-1)}||^2 = (l_{i,j}^{(p)})^2, j \in \mathcal{V}_s^i.
\end{aligned} \tag{25}$$

By denoting

$$\mathbf{Z}_i^{(p)} = \begin{bmatrix} \mathbf{I}_s & \mathbf{x}_i^{(p)} \\ \mathbf{x}_i^{(p)T} & \mathbf{Y}_i \end{bmatrix} \in \mathbb{S}_+^{s+1} \tag{26}$$

problem (25) can be rewritten in a matrix form with the rank constraint

$$\begin{aligned}
\min_{\mathbf{Z}_i^{(p)},\mathbf{l}_i^{(p)}} \quad & \sum_{j\in\mathcal{N}_s} |l_{i,j}^{(p)} - d_{i,j}|^2 \\
\text{s.t.}\ & \mathbf{Z}_{i(1:s,1:s)}^{(p)} = \mathbf{I}_s \\
& [\mathbf{0};\mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z}_i^{(p)} [\mathbf{0};\mathbf{e}_i - \mathbf{e}_j] = l_{i,j}^2,\ \forall j \in \mathcal{V}_s^i \\
& [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z}_i^{(p)} [\mathbf{x}_k; -\mathbf{e}_i] = \hat{d}_{k,i}^2,\ \forall k \in \mathcal{V}_a^i \\
& \mathbf{Z}_i^{(p)} \succeq 0,\ \operatorname{rank}(\mathbf{Z}_i^{(p)}) = s.
\end{aligned} \tag{27}$$

By replacing the rank constraint with its equivalence stated in (7) and considering a weighted penalty term $\langle \mathbf{Z}, \mathbf{W}\rangle$ in the objective function, problem (27) can be solved via the proposed ARMA. Following the procedures of Algorithm 1, the two sequential problems for solving problem (27) are stated ahead. First, given $\mathbf{W}_i^{p-1}$, we will solve

$$\begin{aligned}
\min_{\mathbf{Z}_i^{(p)},\mathbf{l}_i^{(p)}} \quad & \sum_{j\in\mathcal{N}_s^i} |l_{i,j}^{(p)} - d_{i,j}|^2 + \alpha_p \langle \mathbf{Z}_i^p, \mathbf{W}_i^{p-1}\rangle \\
\text{s.t.}\ & \mathbf{Z}_{i(1:s,1:s)}^{(p)} = \mathbf{I}_s \\
& [\mathbf{0};\mathbf{e}_i - \mathbf{e}_j]^T \mathbf{Z}_i^{(p)} [\mathbf{0};\mathbf{e}_i - \mathbf{e}_j] = l_{i,j}^2,\ \forall j \in \mathcal{V}_s^i \\
& [\mathbf{x}_k; -\mathbf{e}_i]^T \mathbf{Z}_i^{(p)} [\mathbf{x}_k; -\mathbf{e}_i] = \hat{d}_{k,i}^2,\ \forall k \in \mathcal{V}_a^i \\
& \mathbf{Z}^{(p)} \succeq 0.
\end{aligned} \tag{28}$$

Next, given $\mathbf{Z}_i^p$, we will determine $\mathbf{W}_i^p$ by solving

$$\begin{aligned}
\min_{\mathbf{W}_i^p} \quad & \sum_{j\in\mathcal{N}_s^i} |l_{i,j}^{(p)} - d_{i,j}| + \alpha_p \langle \mathbf{Z}_i^p, \mathbf{W}_i^p, \rangle \\
\text{s.t.}\ & \operatorname{tr}(\mathbf{W}_i^p) \geq n - r \\
& \mathbf{I} - \mathbf{W}_i^p \succeq 0 \\
& \mathbf{W}_i^p \succeq 0.
\end{aligned} \tag{29}$$

When the objective value in (28) is sufficiently small at the pth step such that

$$\sum_{j\in\mathcal{N}_s^i} |l_{i,j}^{(p)} - d_{i,j}| + \alpha_p \langle \mathbf{Z}_i^p, \mathbf{W}_i^{p-1}\rangle \leq \epsilon_1 \tag{30}$$

then the solution $\mathbf{Z}_i^p$ satisfies all equality constraints and $\operatorname{rank}(\mathbf{Z}_i^p) = s$, which implies the solution is the exact one. To accelerate the convergence, the sensor node i in subproblem (27) will be treated as a new "anchor" and its position will not be updated at the next step $p+1$ if (30) holds at step p. In addition, if any subproblem is infeasible at one iteration, the estimation of the corresponding node will not be updated at the end of that step.

The distributed algorithm is summarized in Algorithm 2, where ϵ_1, ϵ_2, and ϵ_3 are sufficiently small constants, and the initial guesses of the sensors' positions are randomly generated as neither global nor local information being available at the beginning.

Algorithm 2: Distributed ARMA for SNL Problem.

Input: $x_k, k \in \mathcal{V}_a, \mathcal{E}_{ss}, \mathcal{E}_{as}, d_{i,j}, \hat{d}_{k,i}, \epsilon_1, \epsilon_2, \epsilon_3, \theta, \{\alpha\}$.
Output: Local optimum $\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_m^*$.

begin
  Initialize the random locations of sensors: $\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_m^0$, and $\mathbf{X}^{(0)} = [\mathbf{x}_1^0, \cdots, \mathbf{x}_m^0]^T$. Initialize the random $\mathbf{W}_i^0$.
  for $p \leftarrow 0$ to $p_{\max}$ **do**
    for each sensor $i \leftarrow 1$ to m **do**
      update $\mathcal{V}_a^i, \mathcal{V}_s^i$ from $\mathbf{X}^{(p)}$;
      update $\mathbf{Z}_i^{p+1}$ by solving (28) with $\mathbf{W}_i^p$;
      update $\mathbf{W}_i^{(p+1)}$ by solving (29) with $\mathbf{Z}_i^{p+1}$;
      if $\sum_{j\in\mathcal{N}_s^i} |l_{i,j}^{(p)} - d_{i,j}| + \alpha_p \langle \mathbf{Z}_i^p, \mathbf{W}_i^{p-1}\rangle \leq \epsilon_1$ **then**
        Set node i as an anchor;
      end if
      update $\mathbf{x}_i^{(p+1)}$ in (26) from $\mathbf{Z}_i^{(p+1)}$
    end for
    update $\mathbf{X}^{(p+1)} = [\mathbf{x}_1^{(p+1)}, \cdots, \mathbf{x}_m^{(p+1)}]^T$.
    if $\sum_{i=1}^m \langle \mathbf{Z}_i^{(p+1)}, \mathbf{W}_i^{(p+1)}\rangle \leq \epsilon_3$ & $\|\mathbf{X}^{(p+1)} - \mathbf{X}^{(p)}\| \leq \epsilon_2$ **then**
      break;
    else
      $\mathbf{X}^{(p+1)} \leftarrow \mathbf{X}^{(p+1)} + \theta(\mathbf{X}^{(p+1)} - \mathbf{X}^{(p)})$
      $p \leftarrow p + 1$
    end if
  end for
end

V. Simulations

In this section, both noise-free and noisy simulation examples are presented to demonstrate the performance of the proposed ARMA in centralized (Algorithm 1) and distributed (Algorithm 2) frameworks for solving SNL problems. Comparisons of the SDPR in [19] and a sparse version of full SDP (SFSDP) in [43] are also presented. Both frameworks are implemented in MATLAB SeDuMi [40]. The simulations are carried out on a PC with a 3.60 GHz Intel Xeon E5 processor and 32 GB of RAM. In Algorithms 1 and 2, some parameters are set as $\epsilon_1 = 1e-10, \epsilon_2 = 1e-4, \epsilon_3 = 1e-6$. The performance of both frameworks is evaluated by comparing two indexes: 1) rms error Err_{RMS} and 2) maximum error $Err_{\max}$, defined as

$$Err_{\text{RMS}} = \sqrt{\frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_i - \mathbf{x}_i^{\text{real}}\|^2} \tag{31}$$

$$Err_{\max} = \max_{i=1,\dots,m} \|\mathbf{x}_i - \mathbf{x}_i^{\text{real}}\|. \tag{32}$$

A. SNL for an Infinitesimally Rigid Grounded Graph

Consider a simple 2-D case with 3 anchors and 2 sensors shown in Fig. 3, where the three anchors (black diamonds) are located at $(-0.05, -0.08)^T, (0, 0.05)^T, (0.02, -0.05)^T$, blue circles represent the real positions of 2 sensors, and blue crosses

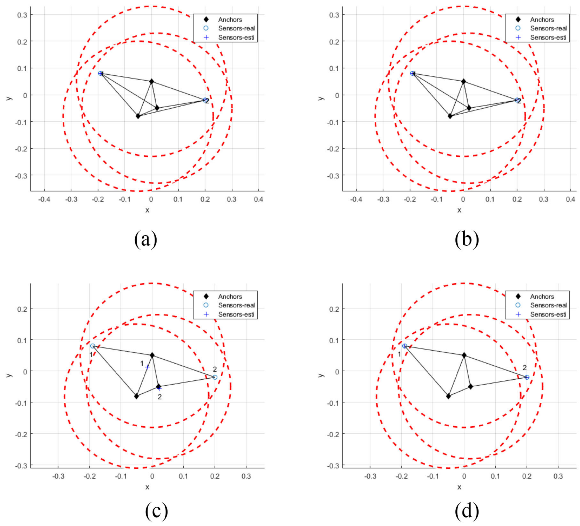


Fig. 3. Comparisons of SDPR and ARMA under different measurement ranges. (a) Results from SDPR with $R_0 = 0.28$. (b) Results from ARMA with $R_0 = 0.28$. (c) Results from SDPR with $R_0 = 0.23$. (d) Results from ARMA with $R_0 = 0.23$.

are the estimated locations. In addition, red-dashed circles show the radio ranges of the anchors. As shown in Fig. 3(a) and (b), when $R_0 = 0.28$, the network is globally rigid, then both SDP relaxation and ARMA can solve the problem exactly. Whereas when $R_0 = 0.23$, the grounded graph is infinitesimally rigid but not globally rigid, then each sensor to be determined has only two distance measurements. Fig. 3(c) demonstrates that the estimations from SDP relaxation are incorrect, as circles and crosses do not overlap. In contrast, by considering the non-adjacency inequality constraints and the rank constraint, the proposed ARMA can obtain the exact solution where circles and crosses overlap, as shown in Fig. 3(d). In fact, the grounded graph in this example satisfies the conditions stated in Theorem 3.8.

B. Randomly Generated SNL Cases

1) SNL in Noise-Free Environments: To further compare the performance of the proposed algorithms with SDPR [19] and SFSDP [43], Algorithms 1 and 2 are implemented to solve several simulation cases that are randomly generated. In these cases, the anchors are evenly distributed in $[-0.5, 0.5]^2$ and all sensors are randomly generated. All results are listed in Table I, where P_{exact} stands for the percentage of sensors that obtain the exact solutions, T_{total} denotes the total computation time of the centralized algorithm, and T_{node} is used to evaluate the average computation time per node in the distributed algorithm. Note that the randomly generated sensors make it possible that there exists some unlocalizable nodes in the network, which leads to the fact that $P_{\text{exact}} < 100\%$ even when the algorithm converges. In addition, when the number of nodes is greater than 100, both SDPR and the centralized ARMA require significantly long computation time to find the solutions. Hence, we only present the result obtained from SFSDP and the distributed algorithm for the case with 100 sensors in the last column of the table.

TABLE I
PERFORMANCE OF SDPR, ALGORITHM 1 AND ALGORITHM 2 FOR ALL FOUR CASES

Case No.		1	2	3	4
	n	18	18	21	24
	m	50	50	80	100
	R_0	0.18	0.17	0.16	0.15
SDPR	Err_{RMS}	0.0167	0.0144	0.0103	—
	$Err_{\max}$	0.0871	0.0588	0.0721	—
	P_{exact}	90%	74%	96%	—
	T_{total} (s)	23.44	22.12	325.5	—
SFSDP	Err_{RMS}	0.0487	0.0382	0.0367	0.0231
	$Err_{\max}$	0.2778	0.1537	0.2111	0.1833
	P_{exact}	90%	80%	91%	93%
	T_{total} (s)	0.4	0.5	0.59	1
Alg. 1	Err_{RMS}	0.0043	0.0081	2.27e-8	—
	$Err_{\max}$	0.0241	0.0433	7.588e-8	—
	P_{exact}	96%	86%	100%	—
	Iterations	2	7	3	—
	T_{total} (s)	42.7	191.8	1506	—
Alg. 2	Err_{RMS}	0.0223	0.0177	0.0245	0.0123
	$Err_{\max}$	0.1364	0.0838	0.1445	0.1104
	P_{exact}	94%	72%	95%	96%
	Iterations	185	194	112	123
	T_{node} (s)	65.8	47.6	39.8	47.4

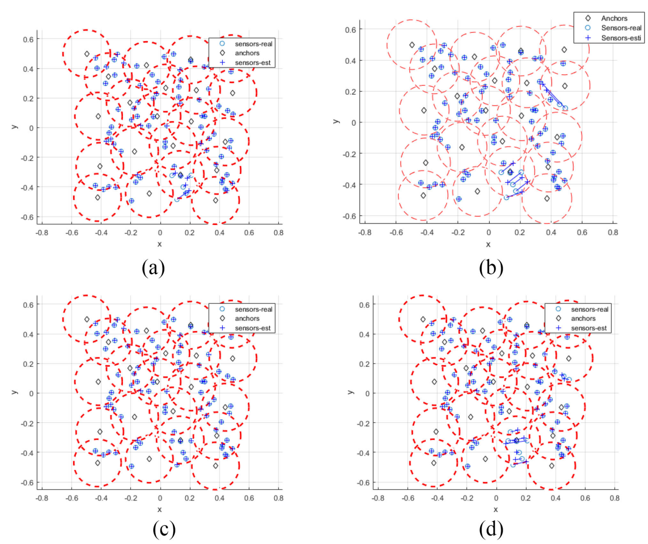


Fig. 4. Comparative results for case 3. (a) Results from SDPR for case 3. (b) Results from SFSDP for case 3. (c) Results from Algorithm 1 for case 3. (d) Results from Algorithm 2 for case 3.

From the comparative simulation results, three conclusions are derived below. First, Err_{RMS} and $\text{Err}_{\max}$ from the centralized ARMA in Algorithm 1 are smaller than those obtained from SDPR. In other words, Algorithm 1 can obtain more precise estimates than SDPR and SFSDP. Meanwhile, the distributed algorithm 2 has similar accuracy when compared to the SDPR and SFSDP. Specifically, Fig. 4 shows the optimized solution from three methods for case 3. One can observe that only the centralized ARMA can perfectly localize all sensors. For SDPR, SFSDP, and distributed algorithm 2 based on random initial guesses, shown in Fig. 4(d), errors exist in the estimations of the same four sensors. The reason why Err_{RMS} and $\text{Err}_{\max}$ from the distributed ARMA are larger than SDPR is that the

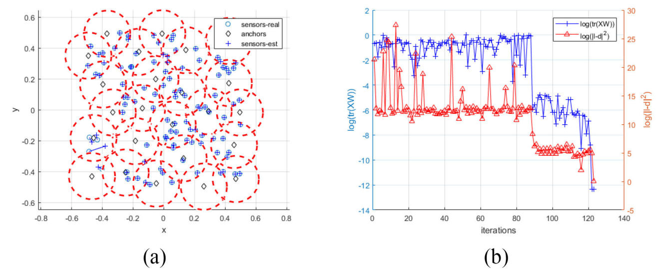


Fig. 5. Results and convergence histories of Algorithm 2 for case 4. (a) Results from Algorithm 2 for case 4. (b) Convergence histories of Algorithm 2 for case 4.

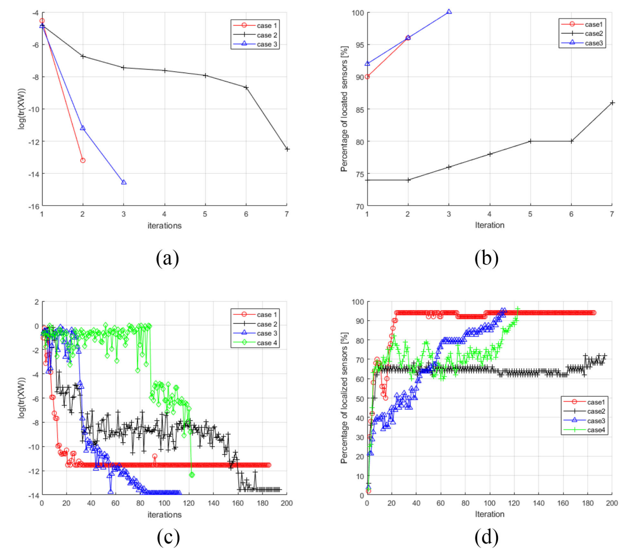


Fig. 6. Convergence histories of Algorithm 1 and 2 for all four cases. (a) Convergence histories of Algorithm 1. (b) Percentage of localized sensors in Algorithm 1. (c) Convergence histories of Algorithm 2. (d) Percentage of localized sensors in Algorithm 2.

converged results of these four nodes in algorithm 2 are incorrect, in fact, these solutions are local optima satisfying all equality constraints.

Second, taking the nonadjacency inequality constraints and rank constraint into account, the centralized ARMA in Algorithm 1 can obtain exact positions for a higher percentage of unknown sensors, which is consistent with Theorem 3.8. Meanwhile, the distributed algorithm can localize the same percentage of sensors compared to the two centralized methods, SDPR and SFSDP. Note that cases 1 and 2 have the same set of anchors but different undetermined sensor sets and radio range. Obviously, a larger radio range leads to a higher percentage of obtained exact solutions from all four methods.

Finally, as the number of nodes increases, the computational time of centralized methods increases much faster than that of the distributed method. For example, the SDPR method takes 22 s to solve a 50-sensor case but 325 s to obtain the optimal solution for an 80-sensor case. Similarly, the centralized ARMA in algorithm 1 takes 2 iterations (42.7 s) in case 1, 7 iterations (192 s) in case 2, and 3 iterations (1506 s) in case 3. However, due to the application of sparsity, the time of SFSDP is much smaller than SDPR and ARMA. With the proposed distributed approach, the computational time per node is about 50 s in all cases, which makes it more scalable for large-scale SNL problems. Fig. 5 presents the results of the proposed distributed algorithm for case 4. As Fig. 5(a) shows, 96% of nodes are determined after 123 iterations. Fig. 5(b) illustrates the convergence history of the distributed method, where the blue-cross curve stands for $\|(\mathrm{tr}(\mathbf{Z}_i^p, \mathbf{W}_i^p))\|_\infty$ and the red-triangle curve represents $\|l^p - d^p))\|_\infty$, where the Y-axis is denoted by $\log_{10}(\|l^p - d^p))\|_\infty)$. When $\|(\mathrm{tr}(\mathbf{Z}_i^p, \mathbf{W}_i^p))\|_\infty$ goes to zero, matrix Z satisfies the rank-s constraint, while $\|l^p - d^p))\|_\infty \to 0$ implies that all equality constraints are gradually satisfied. As shown in Fig. 5(b), the values of $\|(\mathrm{tr}(\mathbf{Z}_i^p, \mathbf{W}_i^p))\|_\infty$ and $\|l^p - d^p))\|_\infty$ dramatically decrease after 90 iterations.

Furthermore, Fig. 6(a) and (c) shows histories of $\mathrm{tr}(\mathbf{Z}, \mathbf{W})$ at each iteration under the proposed centralized and distributed algorithms, which verify the convergence of the proposed ARMA. Fig. 6(b) and (d) illustrates the percentage of localized sensors in Algorithms 1 and 2 for these cases, respectively. Fig. 6(d) shows that most of the sensors are localized within 50 iterations, but it takes more time to localize the rest due to the existence of sensors that are not localizable.

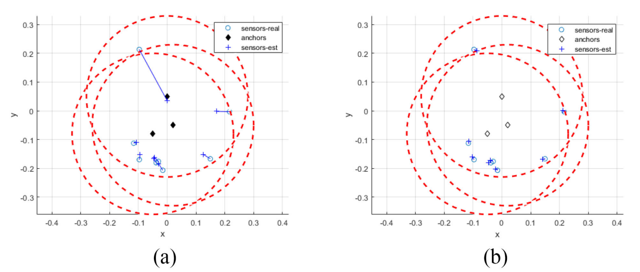


Fig. 7. Comparative results for an SNL with noisy measurements. (a) Localized sensors using SDPR. (b) Localized sensors using Algorithm 1.

2) SNL in Noisy Environments: To evaluate the potential of applying the proposed algorithm to cases with noisy measurements, an SNL with noisy distance measurements is presented here. Let the range measurement be

$$\hat{d}_{i,j} = d_{i,j} + \eta_{i,j}, (i,j) \in \mathcal{E} \tag{33}$$

where $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$ is the true distance between sensors i and j, and $|\eta_{i,j}| \leq \eta_{\max}$ is the unknown and bounded measurement error, respectively. Then, in the SNL problem formulated in (2), all distance constraints are expressed as inequalities

$$d_{ij} - |\eta_{ij}| \leq \|\mathbf{x}_i - \mathbf{x}_j\| \leq \min(d_{ij} + \eta_{ij}, R_0), (i,j) \in \mathcal{E}. \tag{34}$$

Obviously, there is no contradiction between the inequalities in (34) and nonadjacent constraint $\|\mathbf{x}_i - \mathbf{x}_j\| \geq R_0, (i,j) \notin \mathcal{E}_{\mathrm{ss}}$.

In the simulation, let the three anchors (black diamonds) locate at $(-0.05, -0.08)^T, (0, 0.05)^T, (0.02, -0.05)^T$, $R_0 = 0.28$ and $0 \leq \eta_{ij} \leq 0.2 d_{ij}$. Eight sensors to be localized are randomly generated within the box $[-0.25, 0.25]^2$. Fig. 7 presents the localized results from SDPR and the proposed Algorithm 1. It is obvious that the proposed method localizes all of these sensors with much higher accuracy. Accordingly, the errors of Algorithm 1 ($Err_{\text{RMS}} = 0.009$ and $Err_{\max} = 0.0115$) are much smaller than the errors of SDPR ($Err_{\text{RMS}} = 0.0755$ and $Err_{\max} = 0.02024$). Therefore, the proposed centralized method can be extended to noisy cases.

VI. Conclusion

In this paper, the SNL problem has been revisited. We presented novel results on both the graph condition and the position-seeking algorithms. By considering nonadjacency inequality constraints, a milder graph condition for unique localizability has been provided. To solve the distance-based SNL problem, an ARMA has been proposed, which can be applied to a class of general rank-constrained SDP problems. To reduce computational costs and improve scalability, a distributed algorithm based on ARMA has also been proposed. The underlying approach is to decompose SNL into a group of sensor-based subproblems, where each subproblem can be solved by ARMA iteratively using local measurements obtained from its neighbors. Simulation examples in different scales have been presented to demonstrate effectiveness, efficiency, and robustness of the proposed algorithm in centralized and distributed frameworks.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002, doi: 10.1016/S1389-1286(01)00302-4. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128601003024

[2] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments," *Proc. IEEE*, vol. 95, no. 1, pp. 234–254, Jan. 2007, doi: 10.1109/JPROC.2006.887296.

[3] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 36–46, May 2007, doi: 10.1109/MSP.2007.361600.

[4] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, "Environmental wireless sensor networks," *Proc. IEEE*, vol. 98, no. 11, pp. 1903–1917, Nov. 2010, doi: 10.1109/JPROC.2010.2068530.

[5] A. Pawlowski, J. L. Guzman, F. Rodríguez, M. Berenguel, J. Sánchez, and S. Dormido, "Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control," *Sensors*, vol. 9, no. 1, pp. 232–252, 2009, doi: 10.3390/s90100232.

[6] G. Sun, G. Qiao, and B. Xu, "Corrosion monitoring sensor networks with energy harvesting," *IEEE Sensors J.*, vol. 11, no. 6, pp. 1476–1477, Jun. 2011, doi: 10.1109/JSEN.2010.2100041.

[7] T. Sun, L.-J. Chen, C.-C. Han, and M. Gerla, "Reliable sensor networks for planet exploration," in *Proc. IEEE Netw., Sens. Control*, Mar. 2005, pp. 816–821, doi: 10.1109/ICNSC.2005.1461295.

[8] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proc. IEEE*, vol. 95, no. 1, pp. 48–74, Jan. 2007, doi: 10.1109/JPROC.2006.887295.

[9] K. Zhou *et al.*, "Multirobot active target tracking with combinations of relative observations," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 678–695, Aug. 2011, [Online] Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5735231

[10] G. Jing, G. Zhang, H. W. J. Lee, and L. Wang, "Weak rigidity theory and its application to formation stabilization," *SIAM J. Control Optim.*, vol. 56, no. 3, pp. 2248–2273, 2018, doi: 10.1137/17M1122049.

[11] S. Srirangarajan, A. H. Tewfik, and Z. Luo, "Distributed sensor network localization using SOCP relaxation," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4886–4895, Dec. 2008, doi: 10.1109/T-WC.2008.070241.

[12] M. Fanaei, "Distributed detection and estimation in wireless sensor networks," Ph.D. dissertation, Lane Dept. Comput. Sci. Elect. Eng., West Virginia Univ., Morgantown, WV, USA, 2016. [Online] Available: http://community.wvu.edu/mcvalenti/documents/FanaeiDissertation.pdf

[13] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing*. Amsterdam, The Netherlands: Elsevier, 2014, vol. 2, pp. 329–408. doi: 10.1016/B978-0-12-396500-4.00007-7. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123965004000077

[14] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Comput. Netw.*, vol. 51, no. 10, pp. 2529–2553, 2007, doi: 10.1016/j.comnet.2006.11.018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606003227

[15] J. Aspnes, D. Goldenberg, and Y. R. Yang, "On the computational complexity of sensor network localization," in *Algorithmic Aspects of Wireless Sensor Networks*, S. E. Nikoletseas and J. D. P. Rolim, Eds., Berlin, Germany: Springer, 2004, pp. 32–44, doi: 10.1007/978-3-540-27820-7_5.

[16] Z. Zhu, A. M.-C. So, and Y. Ye, "Universal rigidity: Towards accurate and efficient localization of wireless networks," in *Proc. IEEE Conf. Comput. Commun.*, 2010, pp. 1–9, doi: 10.1109/INFCOM.2010.5462057.

[17] B. D. Anderson *et al.*, "Graphical properties of easily localizable sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 177–191, 2009, doi: 10.1007/s11276-007-0034-9.

[18] B. D. Anderson, I. Shames, G. Mao, and B. Fidan, "Formal theory of noisy sensor network localization," *SIAM J. Discrete Math.*, vol. 24, no. 2, pp. 684–698, 2010, doi: 10.1137/100792366.

[19] A. M.-C. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," *Math. Program.*, vol. 109, no. 2, pp. 367–384, Mar. 2007, doi: 10.1007/s10107-006-0040-1. [Online]. Available: https://doi.org/10.1007/s10107-006-0040-1

[20] T. Eren *et al.*, "Rigidity, computation, and randomization in network localization," in *Proc. IEEE 23rd Annu. Joint Conf. Comput. Commun. Soc.*, vol. 4, 2004, pp. 2673–2684, doi: 10.1109/INFCOM.2004.1354686.

[21] J. Aspnes *et al.*, "A theory of network localization," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006, doi: 10.1109/TMC.2006.174.

[22] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Netw.*, vol. 2, no. 2, pp. 188–220, May 2006, doi: 10.1145/1149283.1149286. [Online]. Available: http://doi.acm.org/10.1145/1149283.1149286

[23] N. Ruan and D. Y. Gao, "Global optimal solutions to general sensor network localization problem," *Perform. Eval.*, vol. 75–76, pp. 1–16, 2014, doi: 10.1016/j.peva.2014.02.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531614000303

[24] A. Simonetto and G. Leus, "Distributed maximum likelihood sensor network localization," *IEEE Trans. Signal Process.*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014, doi: 10.1109/TSP.2014.2302746.

[25] X. Zhou, P. Shi, C.-C. Lim, C. Yang, and W. Gui, "A dynamic state transition algorithm with application to sensor network localization," *Neurocomputing*, vol. 273, pp. 237–250, 2018, doi: 10.1016/j.neucom.2017.08.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217313590

[26] A. Stanoev, S. Filiposka, V. In, and L. Kocarev, "Cooperative method for wireless sensor network localization," *Ad Hoc Netw.*, vol. 40, pp. 61–72, 2016, doi: 10.1016/j.adhoc.2016.01.003. [Online]. Available: https://doi.org/10.1016/j.adhoc.2016.01.003

[27] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, "Solving euclidean distance matrix completion problems via semidefinite programming," *Comput. Optim. Appl.*, vol. 12, no. 1, pp. 13–30, Jan. 1999, doi: 10.1023/A:1008655427845. [Online]. Available: https://doi.org/10.1023/A:1008655427845

[28] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, 2004, pp. 46–54, doi: 10.1145/984622.984630. [Online]. Available: http://doi.acm.org/10.1145/984622.984630

[29] P. Tseng, "Second-order cone programming relaxation of sensor network localization," *SIAM J. Optim.*, vol. 18, no. 1, pp. 156–185, 2007, doi: 10.1137/050640308. [Online]. Available: https://doi.org/10.1137/050640308

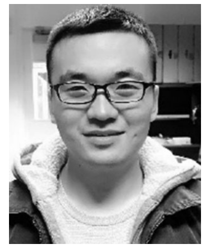
[30] D. Shamsi, N. Taheri, Z. Zhu, and Y. Ye, "Conditions for correct sensor network localization using SDP relaxation," in *Discrete Geometry and Optimization*. New York, NY, USA: Springer, 2013, pp. 279–301, doi: 10.1007/978-3-319-00200-2_16.

[31] Z. Wang, S. Zheng, S. Boyd, and Y. Ye, "Further relaxations of the SDP approach to sensor network localization," *SIAM J. Optim.*, vol. 19, no. 2, pp. 655–673, 2008, doi: 10.1137/060669395

[32] K. W. K. Lui, W.-K. Ma, H.-C. So, and F. K. W. Chan, "Semidefinite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 752–763, Feb. 2009, doi: 10.1109/TSP.2008.2007916.
[33] Q. Shi, C. He, H. Chen, and L. Jiang, "Distributed wireless sensor network localization via sequential greedy optimization algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3328–3340, Jun. 2010, doi: 10.1109/TSP.2010.2045416.
[34] 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, doi: 10.1109/TSP.2014.2339797.
[35] M. Deghat, I. Shames, B. D. Anderson, and J. M. Moura, "Distributed localization via barycentric coordinates: Finite-time convergence*," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 7824–7829, 2011, doi: 10.3182/20110828-6-IT-1002.02448. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016448652
[36] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Netw.*, vol. 2, no. 1, pp. 39–64, 2006, doi: 10.1145/1138127.1138129.
[37] G. C. Calafiore, L. Carlone, and M. Wei, "Distributed optimization techniques for range localization in networked systems," in *Proc. 49th IEEE Conf. Decis. Control*, Dec. 2010, pp. 2221–2226, doi: 10.1109/CDC.2010.5717645.
[38] A. Alfakih, "Graph rigidity via euclidean distance matrices," *Linear Algebra Appl.*, vol. 310, no. 1, pp. 149–165, 2000, doi: 10.1016/S0024-3795(00)00066-5. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379500000665
[39] A. R. Berg and T. Jordán, "Algorithms for graph rigidity and scene analysis," in *Proc. Algorithms—ESA*, G. Di Battista and U. Zwick, Eds., Berlin, Germany: Springer, 2003, pp. 78–89, doi: 10.1007/978-3-540-39658-1_10.
[40] Y. Labit, D. Peaucelle, and D. Henrion, "Sedumi interface 1.02: A tool for solving lmi problems with sedumi," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Des.*, 2002, pp. 272–277, doi: 10.1109/CACSD.2002.1036966.
[41] Q. Li and H.-D. Qi, "A sequential semismooth newton method for the nearest low-rank correlation matrix problem," *SIAM J. Optim.*, vol. 21, no. 4, pp. 1641–1666, 2011, doi: 10.1137/090771181.
[42] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006, doi: 10.1016/j.automatica.2006.02.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0005109806001063
[43] S. Kim, M. Kojima, H. Waki, and M. Yamashita, "Algorithm 920: SFSDP: A sparse version of full semidefinite programming relaxation for sensor network localization problems," *ACM Trans. Math. Softw.*, vol. 38, no. 4, pp. 27:1–27:19, Aug. 2012, doi: 10.1145/2331130.2331135. [Online]. Available: http://doi.acm.org/10.1145/2331130.2331135

Changhuang (Charlie) Wan received the bachelor's and master's degrees in spacecraft design and engineering from Beihang University, Beijing, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree in mechanical and aerospace engineering at the Ohio State University, Columbus, OH, USA.

His research interests include numerical optimization and autonomous systems.



Gangshan Jing received the bachelor's degree in applied mathematics from Ningxia University, Yinchuan, China, in 2012, and the Ph.D. degree in control theory and control engineering from Xidian University, Xi'an, China, in 2018.

He was a Research Assistant with the Department of Applied Mathematics, Hong Kong Polytechnic University, Hong Kong, from December 2016 to May 2017, and November 2017 to January 2018. Since 2018, he has been a Postdoctoral Researcher with the Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH, USA. His current research interests include formation stabilization, distributed optimization, network localization, and motion planning.

Sixiong (Sean) You received the bachelor's degree in automotive engine engineering from Jinglin University, Nanjing, China, in 2014, and the master's degree in mechanical engineering from Tsinghua University, Beijing, China, in 2016. He is currently working toward the Ph.D. degree in mechanical and aerospace engineering at the Ohio State University, Columbus, OH, USA.

His research interests include optimal control and intelligent and autonomous vehicles.

Ran Dai received the bachelor's degree in automation science from Beihang University, Beijing, China, in 2002, and the master's and Ph.D degrees in aerospace engineering from Auburn University, Auburn, AL, USA, in 2005 and 2007, respectively.

She is the Netjets Assistant Professor with the Mechanical and Aerospace Engineering Department, The Ohio State University, Columbus, OH, USA. After graduation, she was an Engineer with the automotive technology company, Dynamic Research, Inc., Torrance, CA, USA, and conducted research and consulting in the areas of semiautonomous vehicle guidance and control. From 2010 to 2012, she was a Postdoctoral Fellow with the Robotics, Aerospace, and Information Networks Laboratory, University of Washington, Seattle, WA, USA, where she was involved in an energy-management project with applications to the next generation of Boeing 787 aircraft power systems. Her research interests include numerical optimization, autonomous system control, and networked dynamic systems.

Dr. Dai is a recipient of the National Science Foundation CAREER Award and NASA Early Faculty Career Award.