

Distributed Event Localization via Alternating Direction Method of Multipliers

Chunlei Zhang and Yongqiang Wang

Abstract—This paper addresses the problem of distributed event localization using noisy range measurements with respect to sensors with known positions. Event localization is fundamental in many wireless sensor network applications such as homeland security, law enforcement, and environmental studies. However, most existing distributed algorithms require the target event to be within the convex hull of the deployed sensors. Based on the alternating direction method of multipliers (ADMM), we propose two scalable distributed algorithms named GS-ADMM and J-ADMM which do not require the target event to be within the convex hull of the deployed sensors. More specifically, the two algorithms can be implemented in a scenario in which the entire sensor network is divided into several clusters with cluster heads collecting measurements within each cluster and exchanging intermediate computation information to achieve localization consistency (consensus) across all clusters. This scenario is important in many applications such as homeland security and law enforcement. Simulation results confirm effectiveness of the proposed algorithms.

Index Terms—Event localization, wireless sensor network, distributed algorithm.



1 INTRODUCTION

WITH the ability to transmit/receive information and fuse data, smart sensors enabled and greatly advanced numerous applications such as environmental monitoring [1], target tracking [2], underwater detection [3], and acoustic gunfire localization [4], [5]. Among these applications, *event localization* is a significant and essential component or even the ultimate goal. Taking the gunfire localization as an example, if some threat sources or impulsive events (e.g., shooting or explosion) occur, it is of imperative importance to localize these threat sources to make prompt reactions (e.g., giving warning, providing aid). In fact, sensor network based event localization has received significant attentions and plenty of techniques have been proposed in the literature, using either angle-of-arrival measurements [6], [7], [8], time-of-arrival (ToA) (including time-difference-of-arrival, i.e., TDoA) measurements [9], [10], or received signal strength (RSS) [11], [12], [13], [14], [15], [16], [17], [18]. There are also some work that discussed the event localization problem based on noisy range measurements directly, which can be obtained based on ToA, TDoA, or RSS information [19], [20], [21], [22], [23]. Generally speaking, these existing methods for event localization formulate the localization problem as a maximum likelihood estimation problem [21] or a least squares problem [17], which is solved by minimizing the non-convex objective function iteratively [11] or by applying various convex relaxations [10].

From the implementation point of view, existing event localization algorithms can be cast into two categories: centralized approaches and distributed approaches. Centralized approaches always gather (noisy) measurements (e.g., range measurements) obtained by all sensors to a processing center, which then estimates the event location using a certain centralized optimization algorithm. Typical centralized methods include the parallel projection method [20], convex relaxation plus semidefinite programming

(SDP) or second-order cone programming method [10], [15], [16], [17], [21], [22], [23]. However, a severe shortcoming of centralized localization algorithms is that the computation complexity at the processing center might be quite high which poses great challenges for low-cost sensor nodes with limited computational capabilities. In addition, the required communication to collect all measurements to a single central node may be problematic due to possible traffic bottleneck and severe constraints on communication ranges. Moreover, once the central node fails due to, e.g., attacks or power depletion, the entire network slips into a state of paralysis. Therefore, techniques solving the event localization problem in a distributed way are crucial for sensor network based event localization.

In contrast to centralized algorithms, distributed localization algorithms are designed to run the computation over the entire network instead of on a processing center. In general, distributed algorithms are often established on massive parallelism or sequential calculations and mutual collaboration [24]. So compared with centralized algorithms, distributed designs have better scalability, flexibility, and failure resilience. One typical distributed approach for event localization is projection-based algorithms which solve the event localization problem by projecting an initial estimate onto sensing disks [11], circles [12], [13], [25], or rings [18]. However, these projection-based localization algorithms are very sensitive to the initial values when the target event lies outside the convex hull of sensors, as will be shown in Sec. 6.

This paper is motivated by acoustic event localization which is crucial on battlefields [26]. In such applications, the target event has no communication or computation capability, which differentiates the problem from sensor localization problems in which the locations of sensors are estimated [27]. Furthermore, in such applications, the target events lie outside the convex hull of deployed sensors, which renders existing projection-based algorithms inappropriate. SDP relaxation based algorithms can avoid the convex hull problem and are traditionally employed to solve the event localization problem [10], [15], [16], [17], [21], [22], [23]. However, as far as we known, existing SDP relaxation

*The work was supported in part by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001.

Chunlei Zhang and Yongqiang Wang are with the department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA {chunlez, yongqiw}@clemson.edu

based algorithms for event localization are all centralized, with a central node collecting and processing all data, which makes them susceptible to processing center failure and traffic bottleneck. In this paper, we propose two distributed event localization approaches based on a clustered architecture motivated by mobile acoustic localization applications such as the PinPoint™ system from BioMimetics Systems Inc. The PinPoint™ mobile localization sensor network can be deployed as a mobile infrastructure for impulsive threat event detection and localization [26], [28]. Each PinPoint™ sensor is a small omnidirectional microphone array which localizes impulsive acoustic events by correlating the ToA measurements among its microphone cells. In fact, since each sensor has an integrated microphone array, individual sensors are able to identify and localize a target event without assistance or cooperation with other sensors. However, due to close distances between the microphone cells, the accuracy of individual sensors is very limited and unsatisfactory, and collaboration among the sensors is necessary to improve localization accuracy [26], [28].

The above application motivated us to assume a localization architecture in which an entire network is divided into several clusters. A cluster head (which can be a regular sensor) collects and fuses measurements (e.g., noisy ranges) obtained from all members in its cluster. Two cluster heads in different clusters can exchange information (the local estimates of target events) if a communication link is available between them; otherwise they don't have access to each other's information. Our developed algorithms can also be applied in some other applications where a cluster-based architecture is employed. A typical example is the wide-area monitoring and control in large-scale power systems [29], [30]. To estimate the electro-mechanical oscillation modes, a large number of phasor measurement units (PMU) have to be deployed across a power network to conduct measurements. The measurements from PMUs have to be fused to diagnose the inter-area oscillation modes. However, wide-area communication between PMUs is very expensive [31]. To fuse information across the PMUs without imposing heavy communication overhead, a similar structure as ours is adopted in [29], [30]. Other examples on cluster-based architecture can be found in [32], [33], [34], [35].

The core of our distributed localization algorithms is the alternating direction method of multipliers (ADMM), which has been proven extremely suitable in distributed convex optimization, especially for large-scale problems [36]. The key idea of ADMM is to obtain a global solution through the cooperation of small local subproblems. ADMM is easy to parallelize and implement, and is robust to noise and computation errors [27]. Our proposed localization approaches take full advantages of ADMM which enables local optimizations within individual clusters as subproblems. Then through cooperation of subproblems in neighboring clusters, a global event localization could be reached. That is to say, the estimated locations obtained by individual clusters are made as consistent as possible. Such consistency is of crucial importance in many applications. For example, when a sporadic impulsive event requiring immediate responsive actions is detected by several monitors, consistency in the estimated location across monitors is the key for multiple monitors to coordinate cooperative operations.

Contribution: The main contribution of this paper is two ADMM-based distributed event localization algorithms, i.e., GS-ADMM and J-ADMM. Compared with existing centralized SDP relaxation based algorithms for event localization, the two algorithms divide the computation on a central node to different

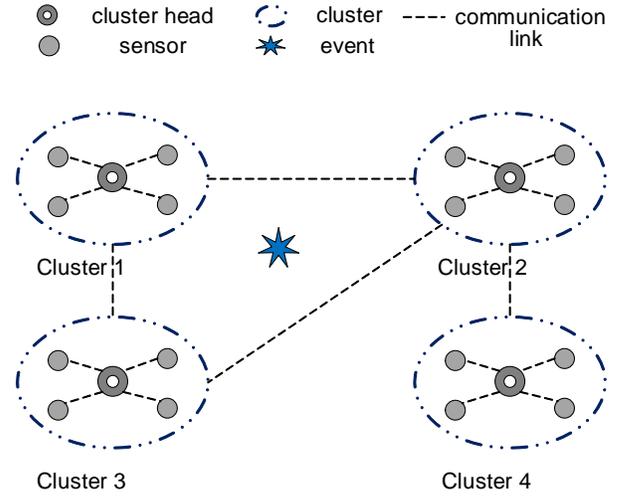


Fig. 1. Cluster based event localization architecture ($m = 4$)

clusters to avoid possible center failure and traffic bottleneck, and in the mean time, guarantee consistency of the estimates across all clusters among which only limited communications are available. Furthermore, the two algorithms take advantages of SDP relaxation to avoid the convex hull problem compared with existing projection-based algorithms. Moreover, the algorithms are proven to converge with a convergence rate of $O(1/t)$ where t is the iteration time.

Organization: The rest of this paper is organized as follows: Section 2 states the formulation of the problem. To solve the problem, a convex relaxation is required and the method proposed by [27] is recapitulated in Section 3. In Section 4, two algorithms named GS-ADMM and J-ADMM are proposed based on ADMM, with their convergence properties analyzed in Section 5. Section 6 gives numerical simulation results. In the end, a conclusion is made in Section 7.

2 PROBLEM STATEMENT

Motivated by mobile acoustic event localization applications such as the PinPoint™ event localization sensor network [26], [28], we consider a localization sensor network divided into m clusters (cf. Fig. 1 for the case $m = 4$). Denote the number of constituent sensors of cluster i as N_i ($i = 1, 2, \dots, m$). We consider localization in D ($D \in \{1, 2, 3\}$) dimensional Euclidean space and suppose that the position of the target event is denoted as $\mathbf{x} \in \mathbb{R}^D$. Denote the position of the k th sensor in the i th cluster as $\mathbf{a}_{i,k} \in \mathbb{R}^D$. The k th sensor in the i th cluster can obtain a noisy range measurement $r_{i,k}$ of its distance with respect to a target event:

$$r_{i,k} = d_{i,k} + v_{i,k}$$

where $d_{i,k} = \|\mathbf{x} - \mathbf{a}_{i,k}\|$ denotes the actual distance between the event position and the k th sensor of the i th cluster, and $v_{i,k}$ is the Gaussian noise term.

Then the event localization problem amounts to estimating the unknown event location \mathbf{x} using known sensor positions $\mathbf{a}_{i,k}$ and noisy range measurements $r_{i,k}$ ($i = 1, 2, \dots, m, k = 1, 2, \dots, N_i$). Still motivated by acoustic event localization applications (e.g., the PinPoint™ event localization sensor network [26], [28]), we assume that a cluster head exists in each cluster

i , which can gather range measurements $r_{i,k}$ from all sensors within the cluster. In addition, a cluster head can communicate and exchange information with the cluster head of a neighboring cluster if there is a communication link between them (cf. Fig. 1). In this case, we also say that these two clusters can communicate. We assume that the communication pattern forms a connected network, i.e., there is a (multi-hop) path (composed of multiple communication links connected in succession) between any pair of cluster heads. For example, in Fig. 1, cluster 1 is able to exchange information with clusters 2 and 3 (via cluster heads); cluster 2 can exchange information with clusters 1, 3, and 4 (via cluster heads), etc. Denote B_i as the set of all neighboring clusters of cluster i , \tilde{B}_i as the union of set B_i and cluster i itself, and $|B_i|$ as the number of clusters in B_i .

As in most existing results, we use the maximum likelihood method for event localization [21], [22]. Let $p_{i,k}(d_{i,k}(\mathbf{x}, \mathbf{a}_{i,k})|r_{i,k})$ denote the measuring probability density function (PDF) for sensor k in cluster i and assume that it is a log-concave function of unknown distance $d_{i,k}$ [27], we can write this problem using the maximum likelihood method (which is costly but efficient [37]):

$$\mathbf{x}_{\text{ML}}^* = \operatorname{argmax}_{\mathbf{x} \in \mathbb{R}^D} \sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k}(\mathbf{x}, \mathbf{a}_{i,k})|r_{i,k}). \quad (1)$$

3 CONVEX RELAXATION

Problem (1) is non-convex and it is generally infeasible to find a global optimal solution [27]. So a convex relaxation is needed to convert problem (1) into a convex optimization problem. Following the idea of [27], we use an SDP based relaxation approach. However, it is worth noting that there are inherent differences between the problem considered here and the sensor-position estimation problem in [27] where each sensor with unknown position estimates its own position using embedded computation capability. The differences are evident from the following example. Suppose that there is only one target to localize. In the case of [27], the target will be a sensor with unknown position and it estimates its own position *alone* using a *centralized* SDP based on all information gathered from adjacent sensors, including their positions and corresponding range measurements. Whereas in our case, the target is an event without any communication or computation capability and the event position estimation process is conducted *cooperatively* in a *distributed* way among the clusters.

To facilitate the relaxation, we first define the following new variables: $y = \mathbf{x}^T \mathbf{x}$, $\epsilon_{i,k} = d_{i,k}^2$. Then we stack $\epsilon_{i,k}, k \in \{1, 2, \dots, N_i\}$ into ϵ_i and further stack $\epsilon_i, i \in \{1, 2, \dots, m\}$ into $\boldsymbol{\epsilon} \triangleq [\epsilon_1^T, \epsilon_2^T, \dots, \epsilon_m^T]^T$. In the same way we stack $d_{i,k}$ into \mathbf{d}_i and $\mathbf{d} \triangleq [\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_m^T]^T$. Then the cost function can be written as

$$f(\mathbf{d}) = - \sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k}|r_{i,k}).$$

Consider the case of white zero-mean Gaussian noise, i.e., $v_{i,k} \sim \mathcal{N}(0, \sigma_{i,k}^2)$, then the above problem can be rewritten as

$$f(\mathbf{d}) = \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k}r_{i,k} + r_{i,k}^2) \quad (2)$$

Without loss of generality, we can set the standard deviation $\sigma_{i,k}$ in (2) to one. Now, problem (1) can be relaxed into the following constrained optimization problem:

$$\begin{aligned} & \min_{\mathbf{x}, \boldsymbol{\epsilon}, \mathbf{d}, y} f(\mathbf{d}) \\ & \text{subject to} \quad y - 2\mathbf{x}^T \mathbf{a}_{i,k} + \|\mathbf{a}_{i,k}\|^2 = \epsilon_{i,k}, \quad y = \mathbf{x}^T \mathbf{x}, \\ & \quad \epsilon_{i,k} = d_{i,k}^2, d_{i,k} \geq 0, \\ & \quad \forall i \in \{1, 2, \dots, m\}, \quad k \in \{1, 2, \dots, N_i\}. \end{aligned} \quad (3)$$

However, in this case, the constraints of (3) still define a non-convex set [27]. Using Schur complements [38], the following convex relaxation can be obtained:

$$\begin{aligned} & \min_{\mathbf{x}, \boldsymbol{\epsilon}, \mathbf{d}, y} f(\mathbf{d}) \\ & \text{subject to} \quad y - 2\mathbf{x}^T \mathbf{a}_{i,k} + \|\mathbf{a}_{i,k}\|^2 = \epsilon_{i,k}, \quad \epsilon_{i,k} \geq 0, \\ & \quad \begin{pmatrix} 1 & d_{i,k} \\ d_{i,k} & \epsilon_{i,k} \end{pmatrix} \succeq 0, \quad d_{i,k} \geq 0, \\ & \quad \forall i \in \{1, 2, \dots, m\}, \quad k \in \{1, 2, \dots, N_i\}, \\ & \quad \begin{pmatrix} \mathbf{I}_D & \mathbf{x} \\ \mathbf{x}^T & y \end{pmatrix} \succeq 0, \quad y \geq 0. \end{aligned} \quad (4)$$

Problem (4) is a convex problem with inequality constraints [36]. We can rewrite the cost function as

$$f(\mathbf{d}, \boldsymbol{\epsilon}) = \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (\epsilon_{i,k} - 2d_{i,k}r_{i,k} + r_{i,k}^2) \quad (5)$$

by enforcing a change of variables $\epsilon_{i,k} = d_{i,k}^2$ to further relax it to a semidefinite programming (SDP) problem [27]. Now, we can propose ADMM based solutions for problem (4).

4 PROPOSED DISTRIBUTED ALGORITHMS

4.1 Preliminaries: Standard ADMM

ADMM is an algorithm which is suitable to solve problems in the following form [36]:

$$\begin{aligned} & \min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ & \text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} = \mathbf{c}. \end{aligned} \quad (6)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{K} \in \mathbb{R}^{p \times m}$, and $\mathbf{c} \in \mathbb{R}^p$, and $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex functions. To get the optimal value $p^* = \inf\{f(\mathbf{x}) + g(\mathbf{z}) \mid \mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} = \mathbf{c}\}$ for problem (6), one can first form an augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}) &= f(\mathbf{x}) + g(\mathbf{z}) \\ &+ \boldsymbol{\mu}^T (\mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} - \mathbf{c}\|^2, \end{aligned}$$

where $\boldsymbol{\mu}$ is the Lagrange multiplier associated with the constraint $\mathbf{A}\mathbf{x} + \mathbf{K}\mathbf{z} = \mathbf{c}$ and $\rho > 0$ is a predefined penalty parameter. Then ADMM solves problem (6) by updating $\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}$ in the following sequence: first an \mathbf{x} -minimization step (7), then a \mathbf{z} -minimization step (8), and finally a dual variable update (9):

$$\mathbf{x}^{k+1} = \operatorname{argmin}_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^k, \boldsymbol{\mu}^k), \quad (7)$$

$$\mathbf{z}^{k+1} = \operatorname{argmin}_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \boldsymbol{\mu}^k), \quad (8)$$

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{K}\mathbf{z}^{k+1} - \mathbf{c}). \quad (9)$$

Next, we will propose two distributed algorithms for event localization using the framework of standard ADMM.

4.2 Problem Reformulation

In distributed algorithms, neighboring nodes have to generate and exchange copies of local estimates to ensure a consistent global estimation across all nodes. In our event localization architecture, a cluster is treated as a normal node which solves a common event localization problem based on measurements obtained by sensors within the cluster. And neighboring clusters exchange intermediate computational results (through cluster heads) to guarantee that all clusters reach the same estimation value.

To better interpret our algorithms, we define a local vector

$$\mathbf{p}_i \triangleq (\boldsymbol{\epsilon}_i^T, \mathbf{d}_i^T, y_i, \mathbf{x}_i^T)^T \in \mathbb{R}^{2N_i+D+1}, \quad i \in \{1, 2, \dots, m\},$$

which is owned by cluster i .

We let \mathbf{p} denote the stacked vector of \mathbf{p}_i and define a convex set

$$\mathcal{P}_i \triangleq \{\mathbf{p}_i | \mathbf{p}_i \text{ verifies (4)}\}.$$

Then problem (4) can be rewritten as

$$\begin{aligned} \min_{\mathbf{p}} \quad & f(\mathbf{p}) \\ \text{subject to} \quad & \mathbf{p}_i \in \mathcal{P}_i, \quad \forall i \in \{1, 2, \dots, m\}, \end{aligned} \quad (10)$$

where, in our situation, $f(\mathbf{p})$ is given as follows:

$$f(\mathbf{p}) = -\sum_{i=1}^m \sum_{k=1}^{N_i} \ln p_{i,k}(d_{i,k} | r_{i,k}) = \sum_{i=1}^m f_i(\mathbf{p}_i). \quad (11)$$

4.3 ADMM based problem formulation

From the architecture in (11), it is easy to see that problem (10) can be divided into m subproblems, which can be solved in a distributed way using ADMM by adding some constraints on \mathbf{p}_i . Next we present the basic idea based on a graph-based formulation of the communication pattern.

Using graph theory [39], the communication pattern of cluster heads can be represented by $G = \{V, E\}$, where the set V denotes the set of cluster heads, and E denotes the set of undirected edges (communication links) between clusters. We use $e_{i,j} \in E, i < j$ to denote the link (if there is) between cluster heads i and j . We use $|E|$ to represent the total number of undirected edges. In our problem formulation, each cluster is associated with a local cost function $f_i(\mathbf{p}_i)$, and all clusters work together to solve the problem in (10). Assume that the local cost function f_i is only known to cluster i , then to reach consistency (consensus) of estimated position values among all clusters, we impose a constraint $\mathbf{x}_i = \mathbf{x}_j$ if there exists an edge $e_{i,j} \in E$ between clusters i and j . Introduce a matrix $J_i = [0_{D \times (2N_i+1)}, I_D] \in \mathbb{R}^{D \times (2N_i+D+1)}$, where I_D denotes the D dimensional identity matrix, then \mathbf{x}_i can be represented as $\mathbf{x}_i = J_i \mathbf{p}_i$. So the constraint $\mathbf{x}_i = \mathbf{x}_j$ can be represented as $J_i \mathbf{p}_i = J_j \mathbf{p}_j$.

Now we are able to rewrite problem (10) into a distributed ADMM form as follows:

$$\begin{aligned} \min_{\mathbf{p}_i, i \in \{1, 2, \dots, m\}} \quad & \sum_{i=1}^m f_i(\mathbf{p}_i) \\ \text{subject to} \quad & J_i \mathbf{p}_i = J_j \mathbf{p}_j, \quad \forall e_{i,j} \in E, \\ & \mathbf{p}_i \in \mathcal{P}_i, \quad \forall i \in \{1, 2, \dots, m\}, \end{aligned} \quad (12)$$

or in a more compact way:

$$\begin{aligned} \min_{\mathbf{p}} \quad & f(\mathbf{p}) \\ \text{subject to} \quad & C J \mathbf{p} = 0, \quad \mathbf{p}_i \in \mathcal{P}_i, \quad \forall i \in \{1, 2, \dots, m\}, \end{aligned} \quad (13)$$

where $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_m^T]^T$, $J = \text{diag}\{J_1, J_2, \dots, J_m\} \in \mathbb{R}^{mD \times (\sum_{i=1}^m 2N_i + D + 1)}$, and C is the edge-node incidence matrix of graph G as defined in [40]. For example, in the one-dimensional case ($D = 1$), $C = [c_{i,j}]$ is an $|E| \times m$ matrix whose $|E|$ rows correspond to the $|E|$ edges and m columns correspond to the m clusters such that:

$$c_{i,j} = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ edge originates at cluster } j, \\ -1 & \text{if the } i^{\text{th}} \text{ edge terminates at cluster } j, \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Here we define that each edge $e_{i,j}$ originates at i and terminates at j .

It can be easily verified that the incidence matrix C for Fig. 1 is

$$C = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}. \quad (15)$$

For high dimensional cases, where $D \geq 2$, $C \in \mathbb{R}^{|E| \times mD}$ can be obtained by replacing the value of 1 and -1 with I_D and $-I_D$, respectively, with I_D denoting the D dimensional identity matrix. Then the C matrix for Fig. 1 becomes

$$C = \begin{bmatrix} I_D & -I_D & 0_D & 0_D \\ 0_D & I_D & -I_D & 0_D \\ I_D & 0_D & -I_D & 0_D \\ 0_D & I_D & 0_D & -I_D \end{bmatrix}. \quad (16)$$

In this formulation, after each cluster obtains its local estimate \mathbf{p}_i , it sends the value $J_i \mathbf{p}_i$ (estimated event position \mathbf{x}_i) to neighboring clusters. By adding the constraint $J_i \mathbf{p}_i = J_j \mathbf{p}_j, \forall i \in \{1, 2, \dots, m\}, j \in B_i$ as shown in (12), the consistency of individual event position $J_i \mathbf{p}_i(\mathbf{x}_i)$ estimated across the clusters is guaranteed. Now we are in place to present our detailed algorithms to solve (12).

Remark 1. Note that although a normal way to apply ADMM to consensus problems is to create auxiliary local variables (cf. [27]), we just put the constraint $J_i \mathbf{p}_i = J_j \mathbf{p}_j$ directly here. The reason that we omit the auxiliary local variables is to save storage space at each cluster, since auxiliary local variables take additional storage space. Furthermore, by adding the constraint $J_i \mathbf{p}_i = J_j \mathbf{p}_j$, we can have both a sequential and a parallel realization with convergence guaranteed, which will be detailed in the following subsection. This kind of constraint and its induced ADMM algorithm is called extended ADMM, which is discussed and applied in many recent work, e.g., [29], [40], [41], [42], [43].

4.4 Proposed Algorithms

Let $\boldsymbol{\lambda}_{i,j}$ be the Lagrange multiplier relevant to the constraint $J_i \mathbf{p}_i = J_j \mathbf{p}_j$. Then the regularized augmented Lagrangian function of problem (12) can be reformulated as

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{p}, \boldsymbol{\lambda}) = & \sum_{i=1}^m f_i(\mathbf{p}_i) \\ & + \sum_{e_{i,j} \in E} (\boldsymbol{\lambda}_{i,j}^T (J_i \mathbf{p}_i - J_j \mathbf{p}_j) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j\|^2), \end{aligned} \quad (17)$$

where $\boldsymbol{\lambda}_{i,j}$ are stacked into $\boldsymbol{\lambda}_i$ for all $j \in B_i$ and $\boldsymbol{\lambda}_i$ are stacked into $\boldsymbol{\lambda}$ for all $i \in \{1, 2, \dots, m\}$.

Applying ADMM, we can get the following two updating recursions:

$$\mathbf{p}^{t+1} = \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}_i} \mathcal{L}_\rho(\mathbf{p}, \boldsymbol{\lambda}^t), \quad (18)$$

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho(J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}). \quad (19)$$

Here, we can update \mathbf{p} in two different ways. One way is based on the Gauss-Seidel update [44] in which clusters update in a sequential order. The other way is the Jacobian scheme in which all clusters update in parallel [45].

Gauss-Seidel update (GS-ADMM): We first consider an algorithm based on the Gauss-Seidel update. Gauss-Seidel update for distributed ADMM has been explored theoretically and proven able to converge in most cases for convex objective functions (see, e.g., [46], [47], [48]). GS-ADMM based solution for distributed event localization can be described as follows:

Algorithm I: GS-ADMM

Each cluster initializes $\mathbf{p}_i^0, \boldsymbol{\lambda}_{i,j}^0$.

Input: $\mathbf{p}_i^t, \boldsymbol{\lambda}_{i,j}^t$

Output: $\mathbf{p}_i^{t+1}, \boldsymbol{\lambda}_{i,j}^{t+1}$

- 1) All clusters update their local vectors in a sequential order and send their local vectors $J_i \mathbf{p}_i^{t+1}$ to neighboring clusters in B_i immediately, where

$$\begin{aligned} \mathbf{p}_i^{t+1} = & \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}_i} f_i(\mathbf{p}_i) + \\ & \sum_{j \in B_i, j \geq i} (\boldsymbol{\lambda}_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^t) + \frac{\rho}{2} \| J_i \mathbf{p}_i - J_j \mathbf{p}_j^t \|^2) + \\ & \sum_{j \in B_i, j < i} (\boldsymbol{\lambda}_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^{t+1}) + \frac{\rho}{2} \| J_i \mathbf{p}_i - J_j \mathbf{p}_j^{t+1} \|^2). \end{aligned} \quad (20)$$

Here we also consider the effect of $J_i \mathbf{p}_i^t$ when updating \mathbf{p}_i^{t+1} by adding a term $\frac{\rho}{2} \| J_i \mathbf{p}_i - J_i \mathbf{p}_i^t \|^2$. Problem (20) with f_i given in (5) is an SDP problem that can be solved by common convex toolboxes such as Yalmip [27], [49], which is used in our simulations.

- 2) Each cluster computes

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho(J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}). \quad (21)$$

- 3) Set $t = t + 1$, and go to 1).
-

In GS-ADMM, all clusters update their local estimated position values in a sequential way just as some projection-based algorithms. Sequential update can be used in small-size networks. For large-scale networks, a parallel method is more appropriate. So we also propose another algorithm based on Jacobian scheme which is amendable for parallelization.

Jacobian based ADMM (J-ADMM): Algorithm J-ADMM is motivated by the work in [42], which proposed the Proximal Jacobian ADMM by adding some proximal terms when updating \mathbf{p}_i . We adopt the same idea here and prove that if the proximal terms meet some additional requirements, convergence of this algorithm can be guaranteed. The detailed procedure of J-ADMM is given as follows, with the convergence analysis detailed in the following section.

Algorithm II: J-ADMM

Each cluster initializes $\mathbf{p}_i^0, \boldsymbol{\lambda}_{i,j}^0$.

Input: $\mathbf{p}_i^t, \boldsymbol{\lambda}_{i,j}^t$

Output: $\mathbf{p}_i^{t+1}, \boldsymbol{\lambda}_{i,j}^{t+1}$

- 1) Each cluster updates its local vector in parallel:

$$\begin{aligned} \mathbf{p}_i^{t+1} = & \operatorname{argmin}_{\mathbf{p}_i \in \mathcal{P}_i} f_i(\mathbf{p}_i) \\ & + \sum_{j \in B_i} (\boldsymbol{\lambda}_{i,j}^{tT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^t) + \frac{\rho}{2} \| J_i \mathbf{p}_i - J_j \mathbf{p}_j^t \|^2) \\ & + \frac{\rho \gamma_i}{2} \| J_i \mathbf{p}_i - J_i \mathbf{p}_i^t \|^2. \end{aligned} \quad (22)$$

The last term of the above equality, i.e., $\frac{\rho \gamma_i}{2} \| J_i \mathbf{p}_i - J_i \mathbf{p}_i^t \|^2$, is the proximal term we added where $\gamma_i \geq 0$ is a scalar. Problem (22) with f_i given in (5) is an SDP problem that can be solved by common convex toolboxes such as Yalmip [27], [49], which is used in our simulations.

- 2) Each cluster sends its local vector $J_i \mathbf{p}_i^{t+1}$ to neighboring clusters in B_i .
- 3) Each cluster computes

$$\boldsymbol{\lambda}_{i,j}^{t+1} = \boldsymbol{\lambda}_{i,j}^t + \rho(J_i \mathbf{p}_i^{t+1} - J_j \mathbf{p}_j^{t+1}). \quad (23)$$

- 4) Set $t = t + 1$, and go to 1).
-

Remark 2. A distinct difference between GS-ADMM and J-ADMM is the way they update \mathbf{p}_i . In GS-ADMM, each cluster updates its local estimated position value in a sequential way, which requires a globally predefined order. Whereas in J-ADMM, all clusters update their local estimated position values simultaneously. We remark that GS-ADMM is appropriate for small-scale sensor networks. But for large-scale networks, updating in a sequential way may be quite time-consuming and parallel methods like J-ADMM are more appropriate. So different updating methods should be chosen according to the size of networks and other practical concerns.

In fact, if we disregard the PinPointTM motivated application scenario, the proposed two algorithms can be completely distributed to each sensor by allowing sensors to have access to neighboring sensors' positions and range measurements with respect to the target event. However, we argue that this, in fact, may cost more energy since each sensor has to solve an SDP problem. In addition, the required storage overhead is larger since each sensor has to store neighboring sensors' positions and range measurements. Furthermore, consider a situation where two sensors can communicate with each other and have the same neighbors. Then the position estimation process conducted at these two sensors are the same, which leads to redundant processing of the same data. While in our clustered architecture, only cluster heads need to conduct position estimation and in fact, each sensor in the cluster can take turns to be the cluster head, which is helpful to average energy consumption. Compared with the iterative schemes, e.g., projection-based algorithms, where each sensor only has access to its own position and range measurement, our algorithms are *insensitive* to the convex hull problem. And compared with centralized SDP-based algorithms, our clustered architecture is robust to processing center failure or traffic bottleneck problems. In addition, the convex relaxation methods used at each cluster can be further improved by using recent works such as [10], [15], [16], [17], [21], [22], [23].

5 CONVERGENCE ANALYSIS

In this section, we analyze the convergence properties of GS-ADMM and J-ADMM. As our algorithms are applications of

distributed ADMM, the analysis benefits from many existing results on general distributed ADMM [40], [41], [50].

5.1 Convergence Analysis of GS-ADMM

Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\boldsymbol{\lambda}_{i,j}^k]_{i,j,e_{i,j} \in E}$ be the iterates generated by algorithm GS-ADMM following (20) and (21). Assume that the initial problem (12) admits a solution $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$, i.e., the Lagrangian function $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$ has a saddle point (note: not the augmented Lagrangian function), then the following theorem holds:

Theorem 1. Let $\bar{\mathbf{p}}^{t+1} = \frac{1}{t+1} \sum_{k=0}^t \mathbf{p}^{k+1}$ be the average of \mathbf{p}^k up to iteration time $t+1$, then the followings hold for all t :

$$(1) \quad 0 \leq L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*) \leq \frac{c_0}{t+1}, \quad (24)$$

(2) The sequence $(\mathbf{p}_1^k, \mathbf{p}_2^k, \dots, \mathbf{p}_m^k)$ deduced by GS-ADMM converge to $(\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_m^*)$, i.e., $\lim_{k \rightarrow \infty} \|\mathbf{p}^k - \mathbf{p}^*\| = 0$. In addition, we have $J_1 \mathbf{p}_1^* = J_2 \mathbf{p}_2^* = \dots = J_m \mathbf{p}_m^*$.

Here

$$c_0 = \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2), \quad (25)$$

and $H = \min\{0, C\}$ ($H_{i,j} = \min\{0, C_{i,j}\}$).

Proof. (24) can be obtained following a way similar to Theorem 4.4 in [40]. A detailed proof is given in Appendix A. To prove the second statement, recall that the objective function is

$$f(\mathbf{d}) = \sum_{i=1}^m \sum_{k=1}^{N_i} \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k} r_{i,k} + r_{i,k}^2).$$

Setting $h_{i,k} = \sigma_{i,k}^{-2} (d_{i,k}^2 - 2d_{i,k} r_{i,k} + r_{i,k}^2)$, we have $f(\mathbf{d}) = \sum_{i=1}^m \sum_{k=1}^{N_i} h_{i,k}$. Note that $h_{i,k}$ is a quadratic function and is strongly convex. Since the sum of strongly convex functions is still strongly convex, our objective function $f(\mathbf{d})$ is strongly convex. Further note that $f(\mathbf{p})$ is equal to $f(\mathbf{d})$ and the set \mathcal{P}_i is convex and closed. Therefore, our problem satisfies the requirements of both strongly convex objective function and convex-and-closed constraint set in [50]. Now we proceed to prove the second statement. First, rewriting $CJ\mathbf{p} = 0$ in the form of $\sum_{i=1}^m [C]_i J_i \mathbf{p}_i = 0$, where $[C]_i$ denotes the columns of C associated with cluster i , we can form a variational inequality $MVI(Q, U)$ similar to (5)-(6) in [50]:

$$\langle \mathbf{u} - \mathbf{u}^*, Q(\mathbf{u}^*) \rangle \geq 0, \quad \forall \mathbf{u} \in U,$$

where

$$\mathbf{u}^* := \begin{pmatrix} \mathbf{p}_1^* \\ \mathbf{p}_2^* \\ \dots \\ \mathbf{p}_m^* \\ \boldsymbol{\lambda}^* \end{pmatrix}, \quad Q(\mathbf{u}^*) := \begin{pmatrix} \boldsymbol{\xi}_1^* + J_1^T [C]_1^T \boldsymbol{\lambda}^* \\ \boldsymbol{\xi}_2^* + J_2^T [C]_2^T \boldsymbol{\lambda}^* \\ \dots \\ \boldsymbol{\xi}_m^* + J_m^T [C]_m^T \boldsymbol{\lambda}^* \\ CJ\mathbf{p} \end{pmatrix},$$

$$U := \prod_{i=1}^m \mathcal{P}_i \times \mathbb{R}^{|E|D}.$$

Then following the proof of Lemma 4.1 in [50], we can get that $(\mathbf{p}_1^{k+1}, \dots, \mathbf{p}_m^{k+1}, \boldsymbol{\lambda}^{k+1})$ is a solution to $MVI(Q, U)$ if $CJ\mathbf{p} = 0$

and $[C]_i J_i \mathbf{p}_i^k = [C]_i J_i \mathbf{p}_i^{k+1}$ hold. Secondly, following the proof of Lemma 4.2 in [50], we can get the following inequality:

$$\begin{aligned} \langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, CJ\mathbf{p} \rangle &\geq \sum_{i=1}^m \omega_i \|\mathbf{p}_i^{k+1} - \mathbf{p}_i^*\|^2 + \rho \|CJ\mathbf{p}^{k+1}\|^2 \\ &+ \rho \sum_{i=1}^m \langle [C]_i J_i \mathbf{p}_i^{k+1} - [C]_i J_i \mathbf{p}_i^*, \sum_{j=i+1}^m ([C]_j J_j \mathbf{p}_j^k - [C]_j J_j \mathbf{p}_j^{k+1}) \rangle \\ &- \rho \sum_{i=1}^m \langle [C]_i J_i \mathbf{p}_i^{k+1} - [C]_i J_i \mathbf{p}_i^*, \frac{1}{|B_i|} ([C]_i J_i \mathbf{p}_i^{k+1} - [C]_i J_i \mathbf{p}_i^k) \rangle, \end{aligned}$$

where $f_i(\mathbf{p}_i)$ is strongly convex with modulus ω_i . Thirdly, define an auxiliary block-diagonal matrix M :

$$M = \begin{pmatrix} \rho m J_1^T [C]_1^T [C]_1 J_1 & \dots & 0 & 0 \\ \dots & \ddots & \dots & \dots \\ 0 & \dots & \rho m J_m^T [C]_m^T [C]_m J_m & 0 \\ 0 & \dots & 0 & \rho^{-1} I \end{pmatrix}.$$

Then by following the idea of the proof of Lemma 4.3 in [50], the following inequality can be obtained:

$$\begin{aligned} \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_M^2 &\leq \|\mathbf{u}^k - \mathbf{u}^*\|_M^2 \\ &- 2 \sum_{i=1}^m \omega_i \|\mathbf{p}_i^{k+1} - \mathbf{p}_i^*\|^2 - \rho \|CJ\mathbf{p}^{k+1}\|^2 \\ &+ 3m\rho \sum_{i=1}^m \|[C]_i J_i \mathbf{p}_i^{k+1} - [C]_i J_i \mathbf{p}_i^*\|^2, \end{aligned}$$

where

$$\|\mathbf{u}\|_M^2 := \|\boldsymbol{\lambda}\|_{\rho^{-1}}^2 + \rho m (\|[C]_1 J_1 \mathbf{p}_1\|^2 + \|[C]_2 J_2 \mathbf{p}_2\|^2 + \dots + \|[C]_m J_m \mathbf{p}_m\|^2).$$

Finally, when $0 < \rho < \min_{1 \leq i \leq m} \left\{ \frac{2\omega_i}{3m\|[C]_i J_i\|^2} \right\}$ holds, we can get the second statement following the proof of Theorem 4.1 in [50]. \blacksquare

Remark 3. Recall $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho CJ\mathbf{p}^{k+1}$, we can get

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \boldsymbol{\lambda}^k + \rho CJ\mathbf{p}^{k+1} \\ &= \boldsymbol{\lambda}^{k-1} + \rho CJ(\mathbf{p}^{k+1} + \mathbf{p}^k) = \dots = \boldsymbol{\lambda}^0 + \rho CJ \sum_{i=1}^{k+1} \mathbf{p}^i. \end{aligned}$$

When $k \rightarrow \infty$, we have $\boldsymbol{\lambda}^{k+1} \rightarrow \boldsymbol{\lambda}^*$. In other words, $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^0 + \rho CJ \sum_{i=1}^{\infty} \mathbf{p}^i$. So c_0 can be represented as:

$$\begin{aligned} c_0 &= \frac{\rho}{2} \|CJ \sum_{i=1}^{\infty} \mathbf{p}^i\|^2 \\ &+ \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2). \end{aligned}$$

It is clear that c_0 will increase with an increase in ρ , so if the iteration time t is fixed, $L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*)$ will also increase with an increase in ρ . That is to say, with ρ increasing, the iteration time to reach convergence will increase, namely convergence rate will be slower. Although with an increase in ρ , the convergence rate will decrease, ρ cannot be too small. This is because if ρ is too small, the constraint $J_i \mathbf{p}_i = J_j \mathbf{p}_j$ is weak, which makes reaching consistency across clusters difficult. More detailed discussions on selecting ρ can be found in [47].

Directly following the statements in Theorem 1, we can obtain the following result on the convergence speed:

Theorem 2. The convergence rate of GS-ADMM is $O(1/t)$, where t is the iteration time.

Proof. The result can be obtained directly from the proof of Theorem 1 and is omitted. ■

5.2 Convergence Analysis of J-ADMM

To analyze the convergence of J-ADMM, we first define several terms: Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\boldsymbol{\lambda}_{i,j}^k]_{i,j,e_i,j \in E}$ be the results for (22) and (23) for iteration k . Augment the coefficients γ_i of proximal terms into a matrix $Q_P = \text{diag}\{\gamma_1 I_D, \gamma_2 I_D, \dots, \gamma_m I_D\}$ and introduce a positive definite diagonal matrix $Q_C = \text{diag}\{|B_1|I_D, |B_2|I_D, \dots, |B_m|I_D\}$, where $|B_i|$ is the number of clusters in B_i . Since Q_C and Q_P are both diagonal matrices, we can define a new diagonal matrix \bar{Q} according to $\bar{Q}^T \bar{Q} = Q_C + I + Q_P$ where I is the identity matrix. It can be easily verified that \bar{Q} has the following form:

$$\bar{Q} = \text{diag}\{\gamma'_1 I_D, \gamma'_2 I_D, \dots, \gamma'_m I_D\}, \quad (26)$$

with $\gamma'_i > 0$ for $i = 1, 2, \dots, m$. Assuming that the original problem (12) admits a solution $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$, then we have the following theorem:

Theorem 3. Let $\bar{\mathbf{p}}^{t+1} = \frac{1}{t+1} \sum_{k=0}^t \mathbf{p}^{k+1}$ be the average of \mathbf{p}^k up to iteration time $t+1$ and denote the eigenvalues of $C^T C$ as α_i . If $\gamma'_i \geq \sqrt{\alpha_{\max}}$ is true with $\alpha_{\max} = \max\{\alpha_i\}$, then the following holds for all t :

$$0 \leq L(\bar{\mathbf{p}}^{t+1}, \boldsymbol{\lambda}^*) - L(\mathbf{p}^*, \boldsymbol{\lambda}^*) \leq \frac{c_1}{t+1}, \quad (27)$$

where $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$ is the Lagrangian function, and

$$c_1 = \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 + \frac{\rho}{2} (\|\bar{Q} J(\mathbf{p}^0 - \mathbf{p}^*)\|^2). \quad (28)$$

Proof. See Appendix B. ■

From Theorem 3, we can easily obtain the following results on the convergence speed:

Theorem 4. The convergence rate of J-ADMM is $O(1/t)$, where t is the iteration time.

Proof. The result can be obtained directly from the proof of Theorem 3 and is omitted. ■

Since c_0 and c_1 are of the same form, Remark 3 for GS-ADMM also applies to the J-ADMM case. Next, we use numerical results to evaluate the performance of GS-ADMM and J-ADMM.

6 SIMULATION RESULTS

In this section, we illustrate effectiveness of the proposed approaches using comparison with existing results. A typical type of distributed algorithms for event localization is the projection-based algorithms. However, some projection-based algorithms, e.g., the DAPA algorithm in [18], is found in our simulations not appropriate for the considered case where the target event lies outside the convex hull of sensors. More specifically, we set the sensor localization architecture similar as in [28], [51], which considers a practical acoustic event localization system (see Fig. 2 for the detailed spatial distribution of all sensor nodes). The target event occurs at $\mathbf{x} = [-5; 200]$, which is far away from the nine sensors. Simulation results suggested that DAPA did not work well in this architecture, even if we set the initial values close to the target

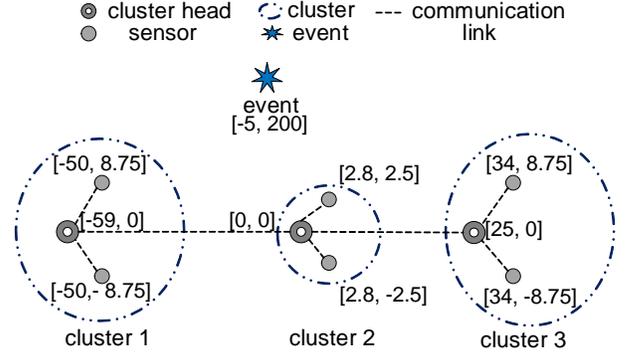


Fig. 2. Event localization architecture used in simulations. The values in $[\bullet]$ denote positions (x, y coordinates) of sensors.

event and used the range measurements without noise, although it did work very well if the target event was set in the convex hull of sensors. In the simulation, we used the same parameters for DAPA as in [18], i.e., $\alpha_1 = \dots = \alpha_9 = \frac{1}{t+2}$, $\beta_1 = \dots = \beta_9 = \frac{1}{t+1}$, $b_1 = \dots = b_9 = 1$, and $\xi_1 = \dots = \xi_9 = 3$.

Then, we compared the localization performance of the proposed algorithms GS-ADMM and J-ADMM with two other projection-based algorithms: the PPM algorithm proposed in [20] and the PONLM algorithm proposed in [12], which gave reasonable performance in the simulations. PPM is a parallel projection method which requires a central node to average the local event location estimates obtained from all sensors in every iteration. PONLM is a sequential projection-based algorithm which solves the event localization problem by finding a point at the intersection of sensing circles. Both localization error (differences between estimated and actual target event positions) and localization consistency (differences in estimated positions between clusters) are compared under different noise standard deviations $\sigma_{i,k}$. The convergence performance is evaluated by exploring the evolution of the localization error with iteration time t .

To facilitate comparison, we first define two performance indices:

Localization Error: we use the root mean square error (RMSE) to quantify the error between estimated and true positions for every cluster or sensor, which is denoted as ERR_{RMSE} :

$$\text{ERR}_{\text{RMSE}} = \sqrt{\frac{\sum_{j=1}^L \|\mathbf{x}_j - \mathbf{x}^*\|^2}{L}},$$

where L is the number of Monte Carlo trials, \mathbf{x}_j is the estimated position in the j th Monte Carlo trial in a certain cluster or sensor, and \mathbf{x}^* is the true position of the target event.

Localization Inconsistency: We also use the root mean square error (RMSE) to quantify the localization inconsistency (difference) in estimated event positions between m clusters, which is denoted as INC_{RMSE} :

$$\text{INC}_{\text{RMSE}} = \sqrt{\frac{\sum_{k=1}^L \sum_{i=1}^{m-1} \sum_{j=i+1}^m \|\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\|^2}{L}},$$

where L is the number of Monte Carlo trials, $\mathbf{x}_{i,k}$ is the estimated position obtained from the i th cluster in the k th Monte Carlo trial. m is the number of clusters.

6.1 Convergence performance

We compared the convergence performance of our sequential GS-ADMM algorithm, parallel J-ADMM algorithm, the sequential PONLM algorithm in [12], and the parallel PPM algorithm in [20]. For GS-ADMM and J-ADMM, we set $\rho = 10^{-3}$. For PPM and PONLM, we set the initial point at $[-50; 100]$ (PPM and PONLM are sensitive to initialization settings, which will be shown later). We used the range measurements without noise in this part. The simulation results are given in Fig. 3.

From Fig. 3, we can see that both GS-ADMM and J-ADMM reached an accuracy of 10^0 after about 10 iterations, while PONLM took 25 iterations and PPM took about 150 iterations. Note that sensors and clusters have to exchange local estimates in each iteration, so the required communication overhead is heavier with an increase in iteration times. The same conclusion can be drawn for energy consumption. It is worth noting that both PPM and PONLM can reach very high accuracies. However, in practical applications like gunfire localization, the accuracy of 10^0 is sufficient [28].

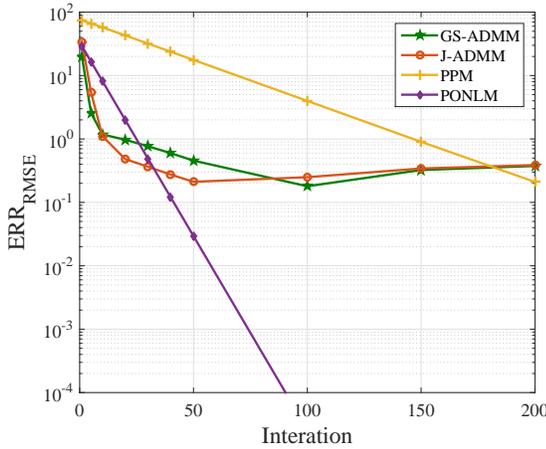


Fig. 3. The evolution of localization error

Remark 4. In our simulations, we used the Sedumi solver in Yalmip, whose limited precision may lead to approximate minima when solving subproblems (20) and (22). This may also lead to a low convergence speed or even fluctuations after a certain number of iterations. In addition, SeDuMi may sometimes return the message “Run into numerical problems”, which implies that it has terminated before it finds an approximate optimal solution [52]. In this situation, we can transform semi-definite inequality constraints into definite inequality constraints by introducing a constant positive definite term (e.g., 10^{-6}) as indicated in [53]. However, such a transformation may bring fluctuations to the convergence process.

6.2 The influence of noise level on ERR_{RMSE}

In this section, we simulated the event localization algorithms under different levels of Gaussian noise standard deviation $\sigma_{i,k}$. For GS-ADMM and J-ADMM, we set $\rho = 10^{-3}$. For PPM and PONLM, we ran simulations under two cases: setting fixed initial values at $[-50; 100]$ (denote as Fix in Table 1) and setting random initial values in the area of $10000m \times 10000m$ (denote as Ran in

Table 1). The number of iterations is fixed to 50 for GS-ADMM, J-ADMM, PONLM, and 200 for PPM. All simulation results are summarized in Table 1 and Fig. 4. Each data point in Table 1 is an average of 100 Monte Carlo trials.

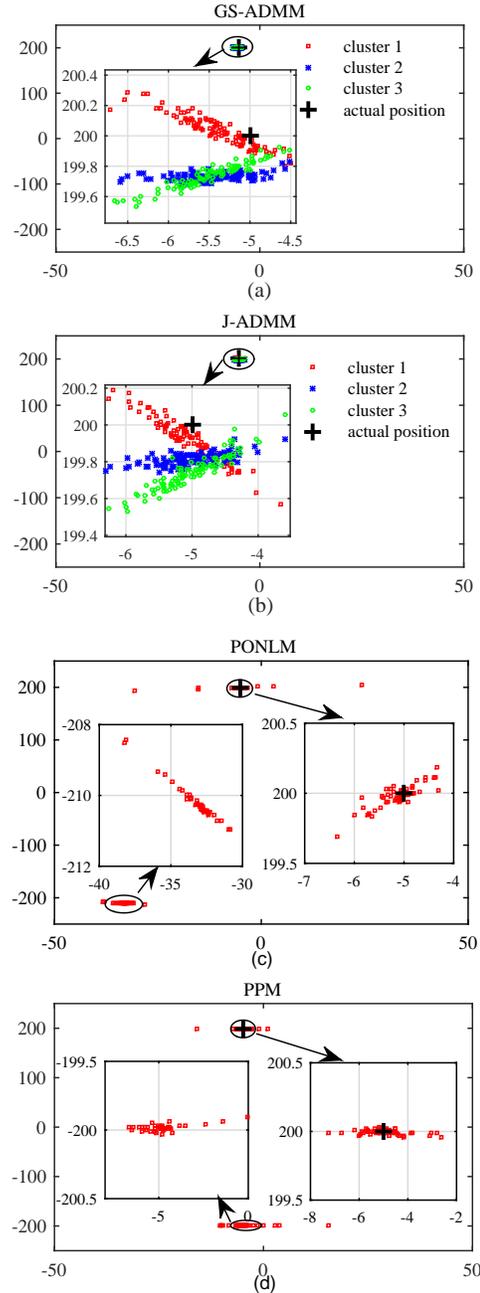


Fig. 4. The distribution of estimated event location, $\sigma = 0.05$. (a) GS-ADMM; (b) J-ADMM; (c) PONLM; (d) PPM.

From Table 1, we can see that both PPM and PONLM reached high localization accuracies under fixed initial values. However, their performance deteriorated significantly when random initial values were used. Therefore PPM and PONLM are sensitive to initial value settings. If the target event lies outside the convex hull of sensors, the convergent values of PPM and PONLM may be far away from the true event position. GS-ADMM and J-ADMM can avoid the convex hull problem, so every estimate lay close to the true event position.

Fig. 4 visualizes the estimated event locations. Fig. 4 (a)

TABLE 1
ERR_{RMSE} of GS-ADMM, J-ADMM, PPM, and PONLM under different measurement noise

$\sigma_{i,k}$	GS-ADMM			J-ADMM			PPM		PONLM	
	CL ₁	CL ₂	CL ₃	CL ₁	CL ₂	CL ₃	Fix	Ran	Fix	Ran
0.00	0.3693	0.4848	0.5128	0.1146	0.2223	0.2964	0.2100	273.19	0.0338	304.52
0.01	0.5417	0.5443	0.5753	0.3546	0.3911	0.4385	0.2106	285.65	0.0498	269.86
0.02	0.5453	0.5862	0.5992	0.2766	0.3266	0.3779	0.2145	282.83	0.0865	307.95
0.05	0.6055	0.6723	0.7188	0.4987	0.5261	0.5724	0.2265	268.31	0.1895	278.84
0.10	1.0564	1.0942	1.1440	1.0562	1.0589	1.1017	0.2832	288.41	0.4019	243.67

and (b) show the localization results of the proposed algorithms GS-ADMM and J-ADMM respectively from 100 Monte Carlo trials with $\rho = 10^{-3}$. Fig. 4 (c) and (d) show the results of PPM and PONLM respectively where the initial positions are chosen randomly. It is clear that both GS-ADMM and J-ADMM performed better than PPM and PONLM when the initial values are randomly chosen.

6.3 The influence of noise level on INC_{RMSE}

Setting $\rho = 10^{-3}$, we also evaluated the influence of noise level on INC_{RMSE} of our proposed algorithms. The results are summarized in Fig. 5.

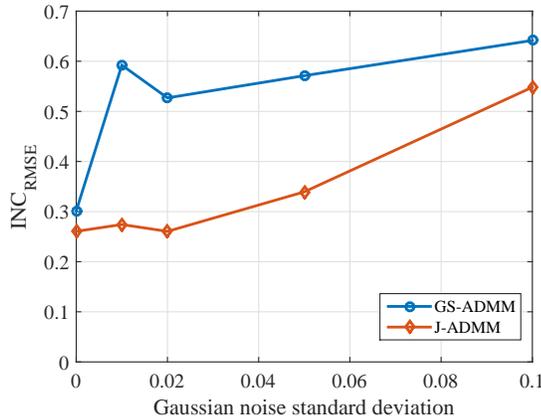


Fig. 5. The influence of measurement noise on localization inconsistency

Fig. 5 indicates that the proposed GS-ADMM and J-ADMM have small localization inconsistency (INC_{RMSE}) under different noise strength. In other words, our proposed algorithms GS-ADMM and J-ADMM can achieve good consistency across clusters even under large noise standard deviations. As indicated before, consistency is of crucial importance in many applications.

7 CONCLUSIONS

We proposed two ADMM based distributed event localization algorithms GS-ADMM and J-ADMM that do not require the target event to be within the convex hull of the deployed sensors. Convergence properties of the algorithms are analyzed theoretically. Numerical simulations showed that the proposed algorithms are robust to measurement noises and insensitive to convex hull problem compared with existing projection-based algorithms.

ACKNOWLEDGMENTS

We would like to thanks Dr. Andrea Simonetto for providing Matlab codes for his paper [27].

APPENDIX A

PROOF OF (24) IN THEOREM 1

To prove (24) in Theorem 1, we first introduce two lemmas:

Lemma 1. Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\boldsymbol{\lambda}_{i,j}^k]_{ij, e_{i,j} \in E}$ be the iterates generated by GS-ADMM following (20) and (21), then the following inequality holds for all k :

$$\begin{aligned} & f(\mathbf{p}) - f(\mathbf{p}^{k+1}) + (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} + \\ & \rho(\mathbf{p} - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0, \\ & \forall \mathbf{p} \in \{[\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_m^T]^T | \mathbf{p}_i \in \mathcal{P}_i, \forall i \in \{1, 2, \dots, m\}\}, \end{aligned} \quad (29)$$

where C is the edge-node incident matrix defined in (15), $H = \min\{0, C\}$, and I is the identity matrix. (In the following, we only consider \mathbf{p} belonging to the set $\{[\mathbf{p}_1^T, \mathbf{p}_2^T, \dots, \mathbf{p}_m^T]^T | \mathbf{p}_i \in \mathcal{P}_i, \forall i \in \{1, 2, \dots, m\}\}$, so we leave out this constraint in the following lemmas and proofs.)

Proof. Denote by g_i the function

$$\begin{aligned} g_i^k(\mathbf{p}_i) &= \sum_{j \in \tilde{B}_i, j \geq i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^k) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^k\|^2) \\ &+ \sum_{j \in \tilde{B}_i, j < i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^{k+1}) + \frac{\rho}{2} \|J_i \mathbf{p}_i - J_j \mathbf{p}_j^{k+1}\|^2). \end{aligned} \quad (30)$$

From the update rule in (20), we know that \mathbf{p}_i^{k+1} is the optimizer of $g_i^k + f_i$ in the closed and convex set \mathcal{P}_i . Since f_i and g_i^k are convex, and g_i^k is differentiable, following the proof of Lemma 3.1 in [54] (which is also mentioned in Lemma 1 in [55]), we can get

$$f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \nabla g_i(\mathbf{p}_i^{k+1}) \geq 0.$$

Substituting $\nabla g_i(\mathbf{p}_i^{k+1})$ with (30), we have

$$\begin{aligned} & f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ & \left(\sum_{j \in \tilde{B}_i, j \geq i} (J_i^T \boldsymbol{\lambda}_{i,j}^k + \rho J_i^T (J_i \mathbf{p}_i^{k+1} - J_j \mathbf{p}_j^k)) \right) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ & \left(\sum_{j \in \tilde{B}_i, j < i} (J_i^T \boldsymbol{\lambda}_{i,j}^k + \rho J_i^T (J_i \mathbf{p}_i^{k+1} - J_j \mathbf{p}_j^{k+1})) \right) \geq 0. \end{aligned}$$

Noting $\boldsymbol{\lambda}_{i,i} = 0$, using (21) leads to

$$\begin{aligned} & f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ & \left(\sum_{j \in B_i} J_i^T \boldsymbol{\lambda}_{i,j}^{k+1} + \sum_{j \in \tilde{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k) \right) \geq 0. \end{aligned}$$

Noting $\lambda_{i,j} = -\lambda_{j,i}$, from the definition of C , we can rewrite the above inequality as

$$\begin{aligned} & f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \cdot \\ & (J_i^T [C]_i^T \boldsymbol{\lambda}^{k+1} + \sum_{j \in \hat{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k)) \geq 0, \end{aligned} \quad (31)$$

here $[C]_i$ denotes the columns of C associated with cluster i .

Summing both sides of (31) over $i = 1, 2, \dots, m$, and noticing that the following two equations hold [40],

$$\begin{aligned} & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T [C]_i^T \boldsymbol{\lambda}^{k+1} = (J\mathbf{p} - J\mathbf{p}^{k+1})^T C^T \boldsymbol{\lambda}^{k+1}, \\ & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T \left(\sum_{j \in \hat{B}_i, j \geq i} \rho J_i^T (J_j \mathbf{p}_j^{k+1} - J_j \mathbf{p}_j^k) \right) \\ & = \rho (J\mathbf{p} - J\mathbf{p}^{k+1})^T [(-C + H)^T H + I] (J\mathbf{p}^{k+1} - J\mathbf{p}^k), \end{aligned}$$

we can get the lemma. \blacksquare

Lemma 2. Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\lambda_{i,j}^k]_{i,j,e_i,j \in E}$ be the iterates generated by GS-ADMM following (20) and (21), then the following equality holds for all k :

$$\begin{aligned} & - (J\mathbf{p}^{k+1})^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \\ & + \rho (J\mathbf{p}^* - J\mathbf{p}^{k+1})^T (H^T H - C^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ & = -\frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2) \\ & - \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & - \frac{\rho}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2) \\ & - \frac{\rho}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2 \\ & - \frac{\rho}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2. \end{aligned} \quad (32)$$

Proof: Since for a scalar a , $a^T = a$ holds, and recall $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho C J \mathbf{p}^{k+1}$, we can get

$$\begin{aligned} & (\mathbf{p}^{k+1})^T J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \\ & = \frac{1}{\rho} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*). \end{aligned} \quad (33)$$

In addition, as $(\mathbf{p}^*, \boldsymbol{\lambda}^*)$ is the saddle point of the Lagrangian function $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$, we have $CJ\mathbf{p}^* = 0$. So we can establish the following relationships using algebraic manipulation:

$$\begin{aligned} & (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) = \frac{1}{2} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \\ & + \frac{1}{2} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2), \end{aligned} \quad (34)$$

$$\begin{aligned} & (\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T I J (\mathbf{p}^{k+1} - \mathbf{p}^k) = \frac{1}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2 \\ & + \frac{1}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2), \end{aligned} \quad (35)$$

$$\begin{aligned} & (\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T H^T H J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ & = \frac{1}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & + \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2, \end{aligned} \quad (36)$$

$$\begin{aligned} & (\mathbf{p}^{k+1} - \mathbf{p}^*)^T J^T C^T H J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ & = \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k)\|^2 + \frac{1}{2\rho^2} \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \\ & - \frac{1}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2. \end{aligned} \quad (37)$$

Then (32) can be proven by plugging equations (33) to (37) into the left part of (32). \blacksquare

Now we proceed to prove Theorem 1. Set $\mathbf{p} = \mathbf{p}^*$ in (29), and recall $CJ\mathbf{p}^* = 0$, then we have

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \mathbf{p}^{(k+1)T} J^T C^T \boldsymbol{\lambda}^{k+1} + \\ & \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0. \end{aligned} \quad (38)$$

Adding and subtracting the term $\boldsymbol{\lambda}^{*T} C J \mathbf{p}^{k+1}$ from the left side of (38), we can get

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) \\ & - \boldsymbol{\lambda}^{*T} C J \mathbf{p}^{k+1} - \mathbf{p}^{(k+1)T} J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) + \\ & \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T H + H^T H + I) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0. \end{aligned}$$

Now by applying (32) into the above inequality, the following inequality can be obtained:

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} C J \mathbf{p}^{k+1} \\ & - \frac{1}{2\rho} (\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 - \|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2) \\ & - \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^*)\|^2 - \|HJ(\mathbf{p}^k - \mathbf{p}^*)\|^2) \\ & - \frac{\rho}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^*\|^2 - \|J\mathbf{p}^k - J\mathbf{p}^*\|^2) \\ & - \frac{\rho}{2} \|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2 \\ & - \frac{\rho}{2} \|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2 \geq 0. \end{aligned}$$

Summing both sides of the inequality over $k = 0, 1, \dots, t$, we can obtain the following result after some re-arrangement:

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} C J \sum_{k=0}^t \mathbf{p}^{k+1} + \frac{\rho}{2} \cdot \\ & (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2) + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 \\ & \geq \sum_{k=0}^t \frac{\rho}{2} (\|HJ(\mathbf{p}^{k+1} - \mathbf{p}^k) - CJ\mathbf{p}^{k+1}\|^2) \\ & + \sum_{k=0}^t \frac{\rho}{2} (\|J\mathbf{p}^{k+1} - J\mathbf{p}^k\|^2) + \frac{1}{2\rho} \|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*\|^2 \\ & + \frac{\rho}{2} (\|HJ(\mathbf{p}^{t+1} - \mathbf{p}^*) + \|J\mathbf{p}^{t+1} - J\mathbf{p}^*\|^2) \geq 0. \end{aligned}$$

In addition, as our function is convex, we have $\sum_{k=0}^t f(\mathbf{p}^{k+1}) \geq (t+1)f(\bar{\mathbf{p}}^{t+1})$, then we can get

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - (t+1)f(\bar{\mathbf{p}}^{t+1}) - (t+1)\boldsymbol{\lambda}^{*T} C J \bar{\mathbf{p}}^{t+1} \\ & + \frac{\rho}{2} (\|HJ(\mathbf{p}^0 - \mathbf{p}^*)\|^2 + \|J\mathbf{p}^0 - J\mathbf{p}^*\|^2) \\ & + \frac{1}{2\rho} \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2 \geq 0. \end{aligned}$$

Dividing both sides by $-(t+1)$ yields

$$\begin{aligned} & f(\bar{\mathbf{p}}^{t+1}) + \boldsymbol{\lambda}^{*T} C J \bar{\mathbf{p}}^{t+1} - f(\mathbf{p}^*) \\ & \leq \frac{\rho}{2(t+1)} (\| H J(\mathbf{p}^0 - \mathbf{p}^*) \|^2 + \| J \mathbf{p}^0 - J \mathbf{p}^* \|^2) \\ & + \frac{1}{(t+1)2\rho} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \|^2. \end{aligned} \quad (39)$$

Combining the above relationship (39) with the Lagrangian function $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$, (24) in Theorem 1 is proven. ■

APPENDIX B PROOF OF THEOREM 3

To prove Theorem 3, we first introduce two lemmas:

Lemma 3. Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\boldsymbol{\lambda}_{i,j}^k]_{i,j \in E, e_{i,j} \in E}$ be the iterates generated by J-ADMM following (22) and (23), then the following inequality holds for all k :

$$\begin{aligned} & f(\mathbf{p}) - f(\mathbf{p}^{k+1}) + (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1} \\ & + \rho (\mathbf{p} - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \geq 0, \end{aligned} \quad (40)$$

where \bar{Q} is defined in (26).

Proof. Denote by g_i the function

$$\begin{aligned} g_i^k(\mathbf{p}_i) & = \sum_{j \in \hat{B}_i} (\boldsymbol{\lambda}_{i,j}^{kT} (J_i \mathbf{p}_i - J_j \mathbf{p}_j^k)) \\ & + \frac{\rho}{2} \| J_i \mathbf{p}_i - J_j \mathbf{p}_j^k \|^2 + \frac{\rho \gamma_i}{2} \| J_i \mathbf{p}_i - J_i \mathbf{p}_i^k \|^2. \end{aligned} \quad (41)$$

Then following the proof of Lemma 1, we can get

$$\begin{aligned} & f_i(\mathbf{p}_i) - f_i(\mathbf{p}_i^{k+1}) + (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \\ & ([C_i^T \boldsymbol{\lambda}^{k+1} + \sum_{j \in \hat{B}_i} \rho J_j (\mathbf{p}_j^{k+1} - \mathbf{p}_j^k) + \rho \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^k)]) \geq 0. \end{aligned} \quad (42)$$

Summing both sides of the above relation over $i = 1, 2, \dots, m$, and noticing that the following two equations hold,

$$\begin{aligned} & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T \rho (\sum_{j \in \hat{B}_i} J_j (\mathbf{p}_j^{k+1} - \mathbf{p}_j^k) + \gamma_i J_i (\mathbf{p}_i^{k+1} - \mathbf{p}_i^k)) \\ & = \rho (\mathbf{p} - \mathbf{p}^{k+1})^T J^T [-C^T C + Q_C + I + Q_P] J (\mathbf{p}^{k+1} - \mathbf{p}^k), \\ & \sum_{i=1}^m (\mathbf{p}_i - \mathbf{p}_i^{k+1})^T J_i^T [C_i^T \boldsymbol{\lambda}^{k+1} = (\mathbf{p} - \mathbf{p}^{k+1})^T J^T C^T \boldsymbol{\lambda}^{k+1}, \end{aligned}$$

we can get the lemma. ■

Lemma 4. Let $\mathbf{p}^k = [\mathbf{p}_1^{kT}, \mathbf{p}_2^{kT}, \dots, \mathbf{p}_m^{kT}]^T$ and $\boldsymbol{\lambda}^k = [\boldsymbol{\lambda}_{i,j}^k]_{i,j \in E, e_{i,j} \in E}$ be the iterates generated by J-ADMM following (22) and (23). Then the following equality holds for all k :

$$\begin{aligned} & -(\mathbf{p}^{k+1})^T J^T C^T (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \\ & + \rho (\mathbf{p}^* - \mathbf{p}^{k+1})^T J^T (-C^T C + \bar{Q}^T \bar{Q}) J (\mathbf{p}^{k+1} - \mathbf{p}^k) \\ & = -\frac{1}{2\rho} (\| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* \|^2 - \| \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^* \|^2) \\ & + \frac{\rho}{2} (\| C J (\mathbf{p}^{k+1} - \mathbf{p}^*) \|^2 - \| C J (\mathbf{p}^k - \mathbf{p}^*) \|^2) \\ & - \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^*) \|^2 - \| \bar{Q} J (\mathbf{p}^k - \mathbf{p}^*) \|^2) \\ & + \frac{\rho}{2} \| C J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 - \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 \\ & - \frac{1}{2\rho} \| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k \|^2. \end{aligned} \quad (43)$$

Proof. The proof is similar to the proof of Lemma 2 and is omitted. ■

Then following the proof of Theorem 1 (setting $\mathbf{p} = \mathbf{p}^*$ in (40) and applying (43)), we can obtain the following inequality:

$$\begin{aligned} & f(\mathbf{p}^*) - f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} C J \mathbf{p}^{k+1} \\ & - \frac{1}{2\rho} (\| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* \|^2 - \| \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^* \|^2) \\ & + \frac{\rho}{2} (\| C J (\mathbf{p}^{k+1} - \mathbf{p}^*) \|^2 - \| C J (\mathbf{p}^k - \mathbf{p}^*) \|^2) \\ & - \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^*) \|^2 - \| \bar{Q} J (\mathbf{p}^k - \mathbf{p}^*) \|^2) \\ & + \frac{\rho}{2} \| C J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 - \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 \\ & - \frac{1}{2\rho} \| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k \|^2 \geq 0. \end{aligned}$$

Summing both sides of the above inequality over $k = 0, 1, \dots, t$, we can get the following result after some rearrangement:

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} C J \sum_{k=0}^t \mathbf{p}^{k+1} \\ & + \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^0 - \mathbf{p}^*) \|^2 + \frac{1}{2\rho} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \|^2 \\ & \geq \frac{\rho}{2} \| C J (\mathbf{p}^0 - \mathbf{p}^*) \|^2 + \frac{1}{2\rho} \| \boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^* \|^2 \\ & + \sum_{k=0}^t \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 - \| C J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2) \\ & + \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{t+1} - \mathbf{p}^*) \|^2 - \| C J (\mathbf{p}^{t+1} - \mathbf{p}^*) \|^2) \\ & + \sum_{k=0}^t \frac{1}{2\rho} \| \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k \|^2 \\ & \geq \sum_{k=0}^t \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{k+1} - \mathbf{p}^k) \|^2 - \| C \|^2 \| J \mathbf{p}^{k+1} - J \mathbf{p}^k \|^2) \\ & + \frac{\rho}{2} (\| \bar{Q} J (\mathbf{p}^{t+1} - \mathbf{p}^*) \|^2 - \| C \|^2 \| J \mathbf{p}^{t+1} - J \mathbf{p}^* \|^2). \end{aligned}$$

Since $\| C \|^2 = \alpha_{\max}$, \bar{Q} is a diagonal matrix with $\gamma_i' \geq \sqrt{\alpha_{\max}}$, we can get that the right hand side of the above inequality is greater than 0, which leads to

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - \sum_{k=0}^t f(\mathbf{p}^{k+1}) - \boldsymbol{\lambda}^{*T} C J \sum_{k=0}^t \mathbf{p}^{k+1} \\ & + \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^0 - \mathbf{p}^*) \|^2 + \frac{1}{2\rho} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \|^2 \geq 0. \end{aligned}$$

In addition, as our function is convex, we have $\sum_{k=0}^t f(\mathbf{p}^{k+1}) \geq (t+1)f(\bar{\mathbf{p}}^{t+1})$ and

$$\begin{aligned} & (t+1)f(\mathbf{p}^*) - (t+1)f(\bar{\mathbf{p}}^{t+1}) - (t+1)\boldsymbol{\lambda}^{*T} C J \bar{\mathbf{p}}^{t+1} \\ & + \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^0 - \mathbf{p}^*) \|^2 + \frac{1}{2\rho} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \|^2 \geq 0. \end{aligned}$$

By dividing both sides by $-(t+1)$, we can obtain

$$\begin{aligned} & f(\bar{\mathbf{p}}^{t+1}) + \boldsymbol{\lambda}^{*T} C J \bar{\mathbf{p}}^{t+1} - f(\mathbf{p}^*) \\ & \leq \frac{1}{t+1} (\frac{1}{2\rho} \| \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \|^2 + \frac{\rho}{2} \| \bar{Q} J (\mathbf{p}^0 - \mathbf{p}^*) \|^2). \end{aligned}$$

Combining the above relationship with the Lagrangian function $L(\mathbf{p}, \boldsymbol{\lambda}) = f(\mathbf{p}) + \boldsymbol{\lambda}^T C J \mathbf{p}$, we can get Theorem 3. ■

REFERENCES

- [1] P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *IEEE Personal Communications*, 7(5):10–15, 2000.
- [2] V. Cevher, M. F. Duarte, and R. G. Baraniuk. Distributed target localization via spatial sparsity. In *Proceedings of the 16th European Signal Processing Conference*, pages 1–5, 2008.
- [3] J. Heidemann, Y. Li, A. Syed, J. Wills, and W. Ye. Underwater sensor networking: Research challenges and potential applications. *Technical Report ISI-TR-2005-603, USC/Information Sciences Institute*, 2005.
- [4] J. George and L. M. Kaplan. Shooter localization using a wireless sensor network of soldier-worn gunfire detection systems. *Journal of Advances in Information Fusion*, 8(1):15–32, 2013.
- [5] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 1–12, 2004.
- [6] L. M. Kaplan, Q. Le, and P. Molnár. Maximum likelihood methods for bearings-only target localization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3001–3004, 2001.
- [7] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986.
- [8] M. Gavish and A. J. Weiss. Performance analysis of bearing-only target location algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 28(3):817–828, 1992.
- [9] K. C. Ho and W. Xu. An accurate algebraic solution for moving source location using TDOA and FDOA measurements. *IEEE Transactions on Signal Processing*, 52(9):2453–2463, 2004.
- [10] K. Yang, G. Wang, and Z. Q. Luo. Efficient convex relaxation methods for robust target localization by a sensor network using time differences of arrivals. *IEEE Transactions on Signal Processing*, 57(7):2775–2784, 2009.
- [11] D. Blatt and A. Hero. Energy-based sensor network source localization via projection onto convex sets. *IEEE Transactions on Signal Processing*, 54(9):3614–3619, 2006.
- [12] Q. Shi and C. He. Distributed source localization via projection onto the nearest local minimum. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2553–2556, 2008.
- [13] J. Wang and P. Regalia. Sensor network localization via boundary projections. In *CISS*, pages 224–229, 2009.
- [14] W. Meng, W. Xiao, L. Xie, and A. Pandharipande. Diffusion based projection method for distributed source localization in wireless sensor networks. In *2011 IEEE Conference on Computer Communications Workshops*, pages 537–542, 2011.
- [15] G. Wang. A semidefinite relaxation method for energy-based source localization in sensor networks. *IEEE Transactions on Vehicular Technology*, 60(5):2293–2301, 2011.
- [16] G. Wang, Y. Li, and R. Wang. New semidefinite relaxation method for acoustic energy-based source localization. *IEEE Sensors Journal*, 13(5):1514–1521, 2013.
- [17] J. Yuan, W. Ai, H. Deng, T. Shuai, and X. Zhao. Exact solution of an approximate weighted least squares estimate of energy-based source localization in sensor networks. *IEEE Transactions on Vehicular Technology*, 64(10):4645–4654, 2015.
- [18] Y. Zhang, Y. Lou, Y. Hong, and L. Xie. Distributed projection-based algorithms for source localization in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 14(6):3131–3142, 2015.
- [19] A. Beck, P. Stoica, and J. Li. Exact and approximate solutions of source localization problems. *IEEE Transactions on Signal Processing*, 56(5):1770–1778, 2008.
- [20] T. Jia and R. M. Buehrer. A set-theoretic approach to collaborative position location for wireless networks. *IEEE Transactions on Mobile Computing*, 10(9):1264–1275, 2011.
- [21] O. Pinar, J. Gomes, J. Xavier, P. Oliveira, et al. A convex relaxation for approximate maximum-likelihood 2D source localization from range measurements. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2698–2701, 2010.
- [22] X. Fu, F. Chan, W. K. Ma, and H. C. So. A complex-valued semidefinite relaxation approach for two-dimensional source localization using distance measurements and imperfect receiver positions. In *2012 IEEE 11th International Conference on Signal Processing*, volume 2, pages 1491–1494, 2012.
- [23] P. Oğuz-Ekim, J. Gomes, J. Xavier, M. Stošić, and P. Oliveira. An angular approach for range-based approximate maximum likelihood source localization through convex relaxation. *IEEE Transactions on Wireless Communications*, 13(7):3951–3964, 2014.
- [24] J. Bachrach and C. Taylor. Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures*, 1, 2005.
- [25] C. L. Wang, D. S. Wu, S. C. Chen, and K. J. Yang. A decentralized positioning scheme based on recursive weighted least squares optimization for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 64(10):4887–4893, 2015.
- [26] S. Deligeorges, G. Cakiades, J. George, Y. Q. Wang, and F. J. Doyle. A mobile self synchronizing smart sensor array for detection and localization of impulsive threat sources. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 351–356, 2015.
- [27] A. Simonetto and G. Leus. Distributed maximum likelihood sensor network localization. *IEEE Transactions on Signal Processing*, 62(6):1424–1437, 2014.
- [28] G. Cakiades, S. Desai, S. Deligeorges, B. E. Buckland, and J. George. Fusion solution for soldier wearable gunfire detection systems. In *SPIE Defense, Security, and Sensing*, pages 838802–838802, 2012.
- [29] S. Nabavi, J. H. Zhang, and A. Chakraborty. Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional PMU-PDC architectures. *IEEE Transactions on Smart Grid*, 6(5):2529–2538, 2015.
- [30] G. B. Giannakis, V. Kekatos, N. Gatsis, S. J. Kim, H. Zhu, and B. F. Wollenberg. Monitoring and optimization for power grids: A signal processing perspective. *IEEE Signal Processing Magazine*, 30(5):107–128, 2013.
- [31] R. Hasan, R. Bobba, and H. Khurana. Analyzing NASPInet data flows. In *Proceedings of IEEE Power Systems Conference and Exposition*, pages 1–6, 2009.
- [32] S. Venkatesan. Coordinating base stations for greater uplink spectral efficiency in a cellular network. In *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–5, 2007.
- [33] A. Papadogiannis, D. Gesbert, and E. Hardouin. A dynamic clustering approach in wireless networks with multi-cell cooperative processing. In *2008 IEEE International Conference on Communications*, pages 4033–4037, 2008.
- [34] J. Zhang, R. Chen, J. G. Andrews, A. Ghosh, and R. W. Heath. Networked MIMO with clustered linear precoding. *IEEE Transactions on Wireless Communications*, 8(4):1910–1921, 2009.
- [35] S. Jain, S. J. Kim, and G. B. Giannakis. Backhaul-constrained multicell cooperation leveraging sparsity and spectral clustering. *IEEE Transactions on Wireless Communications*, 15(2):899–912, 2016.
- [36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [37] A. Beck, M. Teboulle, and Z. Chikishev. Iterative minimization schemes for solving the single source localization problem. *SIAM Journal on Optimization*, 19(3):1397–1416, 2008.
- [38] D. V. Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36:187–295, 1981.
- [39] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 290. Macmillan London, 1976.
- [40] E. Wei and A. Ozdaglar. Distributed alternating direction method of multipliers. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5445–5450, 2012.
- [41] J. Mota, J. Xavier, P. Aguiar, and M. Puschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.
- [42] W. Deng, M. Lai, Z. Peng, and W. Yin. Parallel multi-block ADMM with $O(1/k)$ convergence. *Journal of Scientific Computing*, 71(2):712–736, 2017.
- [43] K. Yuan, Q. Ling, and Z. Tian. Communication-efficient decentralized event monitoring in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8):2198–2207, 2015.
- [44] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.
- [45] Y. Saad. *Iterative methods for sparse linear systems*. 2003.
- [46] B. S. He, M. Tao, and X. M. Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.
- [47] M. Hong and Z. Q. Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1–2):165–199, 2017.
- [48] M. Tao and X. M. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

- [49] J. Löfberg. Yalmip: A toolbox for modeling and optimization in MATLAB. In *Proceedings of IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004.
- [50] D. Han and X. Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155(1):227–238, 2012.
- [51] F. Núñez, Y. Q. Wang, D. Grasing, S. Desai, G. Cakiadesc, and F. J. Doyle. Pulse-coupled time synchronization for distributed acoustic event detection using wireless sensor networks. *Control Engineering Practice*, 60:106–117, 2017.
- [52] H. Waki. How to generate weakly infeasible semidefinite programs via lasserres relaxations for polynomial optimization. *Optimization Letters*, pages 1–14.
- [53] D. Labit and K. Taitz. Users guide for sedumi interface 1.04. 2002.
- [54] B. S. He and X. M. Yuan. On the $o(1/n)$ convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.
- [55] J. Mota, J. Xavier, P. Aguiar, and M. Püschel. A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions. *arXiv preprint arXiv:1112.2295*, 2011.