Distributed Cooperative Localization of Wireless Sensor Networks with Convex Hull Constraint

Shouhong Zhu*, Member, IEEE and Zhiguo Ding, Member, IEEE

Communications and Signal Processing Group
School of Electrical, Electronic and Computer Engineering
Newcastle University, Newcastle upon Tyne, NE1 7RU, UK
E-mail: shouhong.zhu@ieee.org*, z.ding@newcastle.ac.uk
(* indicates the corresponding author)

Abstract

Localization of wireless sensor networks is aimed at determining the positions of all sensors in a network, usually given a few connected anchor nodes’ positions and certain relative measurements, where the latter could be pairwise distance measurements among directly connected neighbors as considered in this paper. In this paper we investigate the neighborhood collaboration based distributed cooperative localization of all sensors in a particular network with the so-called ‘convex hull constraint’: all nodes in such a network are either connected position-known anchors or sensors to be localized, and every sensor is inside the convex hull of its neighbors. For such a practically widely seen thus important class of localizable wireless sensor networks, we propose three iterative self-positioning algorithms, for independent implementation at all individual sensors of the considered network. Analysis and simulation study show that when iteratively running at all sensors of the considered network, i) the first one of our proposed iterative self-positioning algorithms leads to global convergence, where the converged solution is the correct positions of all sensors in the absence of measurement error, but might not be optimum if there exist measurement errors; ii) the second algorithm suffers from local convergence, but once correctly converged the converged solution would be the least squares (LS) solution; iii) the third algorithm, a combined version of the former two algorithms that switches between their iterations efficiently and independently at individual sensors only based on locally collected information, globally converges to the LS solution as long as the measurement errors are sufficiently small such that the converged solution by the first algorithm is well inside the correctly converging area of the second algorithm.

Index Terms

Wireless sensor network, distributed cooperative sensor localization, neighborhood collaboration, convex hull

The authors were supported by the UK EPSRC research project under grant number EP/F062079/1.
I. INTRODUCTION

Sensor localization is crucial to the potential location-aware applications of wireless sensor networks. For example, the sensors in the wireless sensor network for environment monitoring need to be accurately oriented or localized in order to sense, report, and process relevant environmental events [1]. For a practically randomly deployed wireless sensor network, the localization of its sensors is generally challenging due to the following reasons. Firstly, it is unlikely for every sensor in the network to be equipped with an external self-positioning device, such as a Global Positioning System (GPS) receiver, for its localization due to either feasibility or cost. Secondly, it is also not likely for every sensor in the network to have sufficient position-known anchor nodes within its direct communication and measurement radius, where sensors might have to go more hops to reach an anchor. As a result, cooperative localization is needed, where collaboration among directly connected neighbor sensors must be exploited [2][3]. In addition, the scalability of the network makes many existing cooperative localization approaches relying on centralized implementation prone to severe communication congestion throughout the network and high computational complexity at individual sensors, motivating the use of decentralized or distributed approaches [4][5].

Generally speaking, for a wireless sensor network with arbitrary node deployment and pairwise direct communication and measurement scenario it is difficult to find its corresponding accurate (for the case without measurement error) or optimum (for the case with measurement errors) localization solution [6], and it is more so to find such a localization solution in a practically preferred cooperative and distributed way. In this paper we investigate how to determine, based on the positions of a few connected anchors as well as pairwise distance measurements among neighbors and in a neighborhood collaboration based cooperative and distributed way, the positions of all sensors in a particularly important wireless sensor network with the so-called ‘convex hull constraint’: All nodes in such a network are either connected position-known anchors or sensors to be localized, and every sensor is inside the convex hull of its neighbors. In fact, the wireless sensor network with this convex hull constraint belongs to a practically widely seen thus important class of localizable networks: 1. Such a network constraint is practically feasible and easy to be realized [7][8]. It is easy to deploy a sufficient number of connected anchors at the outskirts of these sensors to ensure all sensors to be inside the convex hull of the anchors, and further all sensors adjust their direct communication and measurement radius to satisfy that every sensor is inside the convex hull of its neighbors. If there exist barriers in the field of these sensors, it is still feasible to keep the convex hull constraint by deploying additional anchors around each of these barriers. 2. The network with this convex hull constraint is localizable, as its corresponding graph (if regarding the nodes as its vertices and neighbor pairs as its edges) is globally rigid. In other words, the network with this convex hull constraint has only one realization given pairwise distances among neighbors [6]. We consider
such an important class of localizable networks, as there exist simple iterative self-positioning algorithms for parallel and independent implementation at all individual sensors of the considered network that can globally converge to the least squares (LS) solution, as will be seen later in this paper.

For our considered wireless sensor network, there are several existing localization schemes that can be applied. We briefly list these related work as below:

The multidimensional scaling (MDS) algorithm can be directly applied if all pairwise distance measurements within the network are available [9]. This is a centralized method that has a closed form localization solution. In the case partial pairwise distance measurements are not available, the MDS algorithm still can be applied by replacing these unavailable measurements with their corresponding shortest paths, as proposed in [10], and subsequently applied iteratively by utilizing the updated sensor position estimates. However, the MDS algorithm and its variants usually have a significant gap in estimation performance as compared with the optimum solution, and needs a subsequent iterative refinement to achieve optimum estimation performance. Further, as mentioned earlier, centralized implementation generally results in communication congestion and high computational complexity with the network scaling, thus not practically preferable.

Information inference based localization schemes, such as the one based on belief propagation (i.e. a posteriori probability distribution dissemination and fusion among neighbors) can be applied. These localization schemes are inherently iterative, and the iterations are based on locally collected information. So these localization schemes are suitable for distributed implementation. However, these localization schemes have problematic convergence due to the loops existing in the network, and also have prohibited computational complexity due to the large parameter (sensor position) space especially when sensors have to go more hops to reach an anchor. Although the later proposed particle based versions [11] can reduce some computational burden at the cost of slow convergence, the reduced computation amount is still unacceptable for the case sensors have to go more than two hops to reach an anchor. Many existing approaches request a priori information on sensor position to reduce parameter space.

Multilateration based incremental localization schemes as in [12] and map stitching based localization schemes as in [13] can be implemented in a centralized way as well as in a distributed way. Their main drawback is that these schemes require more densely deployed sensors with respect to the sensors’ communication and measurement radium to localize additional sensor in an incremental way or form local maps, and such requirement might not be satisfied in our considered network. Furthermore, the obtained localization solutions of these methods still have significant gaps in estimation performance as compared with the optimum solution, and also needs a subsequent iterative refinement to achieve optimum estimation performance.
The LS based search algorithms are also iterative for which the corresponding iterations are based on locally collected information, thus suitable for distributed implementation. In fact the LS based iterative refinement is usually adopted by the localization schemes that can not achieve the CRLB in the case with zero-mean Gaussian distributed measurement errors. However, the global convergence of these algorithms generally can not be guaranteed due to the non-convexity of its cost function [4]. To achieve global convergence, many relaxation based methods have been proposed to balance the tradeoff between global convergence and converged estimation performance. Programming relaxation based localization schemes and algorithms are such examples, which were proposed originally for centralized implementation [14]-[18] and have the potential to be converted for distributed implementation [5][19]. But their global convergence is still not guaranteed theoretically although can be shown numerically. Further, their implementations have high computational complexity.

In [7][20] the authors considered the wireless sensor networks with the above mentioned convex hull constraint, and proposed a distributed cooperative localization scheme as well as its corresponding iterative self-positioning algorithms based on their calculated barycentric coordinates. These algorithms can lead to global convergence. However, in order to calculate barycentric coordinates, these algorithms require the measurements on the mutual distances among the triangulation-chosen neighbors of every sensor, which might not be available in our considered network.

In [8], the authors proposed a distributed cooperative localization scheme and its corresponding iterative self-positioning algorithms. Their proposed scheme and algorithms ensure the localization of the sensors qualifying the convex hull constraint to be globally convergent as long as the updating step size is sufficiently small, but the converged solution suffers from significant gap in estimation performance as compared to the optimum solution. The authors there also proposed several refinement procedures based on various cost functions to be switched after convergence to further improve the converged estimation performance, but the switching operation applied, the cost functions adopted, as well as the updating step size used still make the resultant converged estimation performance far from satisfied.

From the above literature review we can see that typically the distributed cooperative localization of all sensors in a network requests an iterative self-positioning algorithm to be implemented at all sensors. And the requirement on such an iterative algorithm is that its parallel and independent implementation at all sensors of the network must lead to a globally convergent and accurate (in absence of measurement error) or optimum (in presence of measurement errors) localization solution. In this paper, starting from the concept of the global convex cost function as in [8], we propose a distributed cooperative localization scheme and three iterative self-positioning algorithms, all performing simple neighborhood cooperation, for independent implementation at all individual sensors of the considered network.
Our contributions in this paper can be listed as below:

1) We propose a ‘pulled-only’ iterative self-positioning algorithm. When iteratively running at all sensors of the considered network, this algorithm leads to global convergence in the sense of the globally convex cost it minimizes. The converged localization solution is the correct positions of all sensors in the absence of measurement error, but might not be optimum if there exist measurement errors. If the measurement errors are small, the converged solution would be close to the correct positions of all sensors thus the LS solution (because three of them are close to each other when the measurement errors are small).

2) We propose a ‘pulled-or-pushed’ iterative self-positioning algorithm. When iteratively running at all sensors of the considered network, this algorithm suffers from local convergence. But once correctly converged the converged solution would be the LS solution.

3) We propose a combined version of the former two algorithms, which efficiently switches between their iterations. The switches are performed independently at individual sensors based on locally collected information. When iteratively running at all sensors of the considered network, this combined algorithm globally converges to the LS solution as long as the measurement errors are sufficiently small such that the converged solution by the first algorithm is well inside the correctly converging area of the second algorithm.

The rest of this paper is arranged as follows: In Section II, the data model and localization performance measure are given. In Section III, our proposed distributed cooperative localization scheme, including three iterative self-positioning algorithms, with analysis are presented. In Section IV, simulation study is conducted to evaluate our proposed algorithms. And finally Section V concludes the paper.

II. DATA MODEL AND LOCALIZATION PERFORMANCE MEASURE

As the extension to higher dimension cases is straightforward, in this paper, without loss of generality, we only consider 2-D cases, where the location/position can be denoted by a two-element column vector that composes of the two coordinates within a coordinate system.

We consider a wireless sensor network containing $M = M_u + M_a$ nodes with positions $p_m$, $m = 1, \ldots, M$, where $M_u$ the number of the sensors to be localized and $M_a$ is the number of the position-known anchors. The network satisfies the convex hull constraint, i.e. every sensor is inside the convex hull of its neighbors. If node $i$ and node $j$ are neighbors, their pairwise distance measurement $r_{ij} = r_{ji}$ is available and can be represented as

$$r_{ij} = \|p_i - p_j\| + n_{ij}$$
where $\| \cdot \|$ is the norm-2 operator, and $n_{ij} = n_{ji}$ is the corresponding measurement error. The statistics on this measurement error depends on the ranging technique used [21]. Here we simply assume that the measurement errors in all pairwise distance measurements are independent and identically distributed (i.i.d.) zero-mean Gaussian distributed, with common but unknown variance $\sigma_n^2$. The extension to other variance models is straightforward by applying suitable weights in the corresponding processing, and is not included in this paper due to limited space.

For the localization of all sensors of the considered network, a suitable performance measure is the root of average mean squared (r.a.m.s.) positioning error, where ‘averaging’ is carried out over all sensors. If the estimates of the positions of all sensors are $\hat{p}_m$, $m = 1, ..., M_u$, then the corresponding r.a.m.s. positioning error is

$$\text{r.a.m.s. positioning error} = \sqrt{\frac{1}{M_u} \sum_{m=1}^{M_u} E \{ \| \hat{p}_m - p_m \|^2 \}}.$$  

(2)

For the average mean squared (a.m.s.) positioning error, there exists a Cramer-Rao lower bound (CRLB), which can be derived based on the probability density function (pdf) of the available pairwise distance measurements given the positions of all nodes [2][22][23][24][25]. We list its concise form in Appendix A.

### III. Proposed Distributed Cooperative Localization Scheme with Analysis

In this section, we firstly present the procedure and protocol of the scheme, then present three iterative self-positioning algorithms with performance analysis, and finally present the initialization techniques improving convergence speed. We also discuss some relevant issues.

#### A. Procedure and Protocol of Proposed Scheme

The procedure and protocol of our proposed distributed cooperative localization scheme can be listed as below:

1) All nodes in the network work independently.
2) All anchors broadcast their a priori known positions to their neighbors respectively.
3) All sensors broadcast their current (either initialized or updated) estimates on their own positions to their neighbors respectively.
4) All sensors implement common iterative self-positioning algorithm, i.e. every sensor updates its estimate on its own position based on locally collected information. Such locally collected information includes the sensor’s current estimate on its own position, its pairwise distance measurements from its neighbors, and its received data from its neighbors that contains its neighbors’ positions or current estimates on their positions.
In this paper we do not consider the case with communication failures, as very much short communication failures between neighbors only slightly affect the convergence speed whereas very much long communication failures may change the corresponding network graph thus network localizability.

Here we would also like to point that the data exchange between neighbors might be able to produce new measurements on the inter-neighbor distances, but due to limited space in this paper we do not include the investigations that utilize these new measurements for further improving localization performance.

B. Proposed Iterative Self-Positioning Algorithms

1) 'Pulled-Only' Algorithm – Algorithm I: At a sensor, in each iteration of this iterative self-positioning algorithm, the current estimate on the position of this sensor is updated according to its neighbors to produce same number of new estimates. When updated according to a neighbor, the current estimate on the position of the sensor i) stays unchanged if the calculated distance, i.e. the distance between the current estimate on the position of the sensor and the position or the estimate of the position of this neighbor, is no farther than the measured distance, i.e. the obtained measurement on the distance between the sensor and the neighbor; ii) otherwise is pulled by the neighbor to a position such that the newly calculated distance equals to the measured distance. If denoting the current estimate on the position of sensor $i$ at time $n$ as $\hat{p}_i(n)$, we can have the new estimate according to its $m$-th neighbor as

$$
\hat{p}_i^{(m)}(n+1) = \begin{cases} 
\hat{p}_i(n), & \text{if } \|\hat{p}_i(n) - q_m(n)\| \leq r_{im} \\
\hat{p}_i(n) + \frac{q_m(n) - \hat{p}_i(n)}{\|q_m(n) - \hat{p}_i(n)\|} (\|q_m(n) - \hat{p}_i(n)\| - r_{im}), & \text{otherwise}
\end{cases}
$$

(3)

where $q_m(n)$ is position $p_m(n)$ (for anchor) or position estimate $\hat{p}_m(n)$ (for other sensor) of the $m$-th neighbor of sensor $i$ at time $n$. By averaging these new estimates according to all its neighbors, the updated estimate on the position of the sensor is then

$$
\hat{p}_i(n+1) = \frac{1}{M_i} \sum_{m \in S_i} \hat{p}_i^{(m)}(n+1)
$$

(4)

where $M_i$ is the size of set $S_i$, and $S_i$ is the set containing all neighbors of sensor $i$. We can see that the updating operation at each sensor in each iteration is very much simple.

Below we analyze the convergence property as well as the converged estimation performance of the above iterative self-positioning algorithm when iteratively running at all sensors of the considered network.

At any sensor, say sensor $i$, in any iteration, say the one at time $n$, the updating of Algorithm I defined by (3) and (4) can be regarded as the following one-step gradient search in minimizing the corresponding
cost function at a constant step size
\[ \hat{p}_i(n + 1) = \hat{p}_i(n) - \mu \frac{\partial \sum_{m \in S_i} g(p_i, q_m)}{\partial p_i} |_{p_i=\hat{p}_i(n), q_m=q_m(n), m \in S_i} \] (5)

where
\[ g(p_i, q_m) = \begin{cases} 0, & \text{if } \|p_i - q_m\| \leq r_{im} \\ (\|p_i - q_m\| - r_{im})^2, & \text{otherwise.} \end{cases} \] (6)

In such an equivalence, \( g(p_i, q_m) \) is a convex function with respect to (w.r.t.) \( p_i \) if \( q_m \) is fixed (in 2-D the corresponding cost surface is like a convex basin with a round area in the middle having zero-cost). Then if \( q_m, m \in S_i \) all are fixed, \( \sum_{m \in S_i} g(p_i, q_m) \), the sum of several/limited convex functions w.r.t. \( p_i \), is still a convex function w.r.t. \( p_i \).

Further the updating at any sensor in any iteration also can be regarded as the following one-step gradient searching minimizing the following total cost function at a constant step size
\[ \hat{p}_i(n + 1) = \hat{p}_i(n) - \mu \frac{\partial h(p_m, m = 1, \ldots, M_u)}{\partial p_i} |_{p_i=\hat{p}_i(n), p_m=\hat{p}_m(n), m \in S_i} \] (7)

where
\[ h(p_m, m = 1, \ldots, M_u) = \sum_{i=1}^{M_u} \sum_{k=M_u+1}^{M} v_{ik} g(p_i, p_k) + \sum_{i<j, i,j=1}^{M_u} v_{ij} g(p_i, p_j). \] (8)

Here \( v_{ij} = v_{ji} \) is 1 if node \( i \) and node \( j \) are a pair of neighbors and their pairwise distance measurement is available, and is 0 otherwise.

From such an equivalence, we have the following theorem for the (global) convexity of this total cost function w.r.t. the composed vector \([\hat{p}_1^T, \ldots, \hat{p}_{M_u}^T]^T\) if \( p_k, k = M_u + 1, \ldots, M \) all are fixed.

**Theorem 1.** Given a network containing sensors with unknown positions \( p_m, m = 1, \ldots, M_u \) and anchors with known positions \( p_m, m = M_u + 1, \ldots, M \) as well as certain pairwise distance measurements, the total cost function defined by (8) and (6) is (globally) convex w.r.t. \([\hat{p}_1^T, \ldots, \hat{p}_{M_u}^T]^T\), the vector composed of the positions of all sensors.

**Proof:** See Appendix B.

We also have the following theorem for the monotonically decreasing of this total cost during the proposed gradient search in Algorithm I.

**Theorem 2.** Given a network containing sensors with unknown positions \( p_m, m = 1, \ldots, M_u \) and anchors with known positions \( p_m, m = M_u + 1, \ldots, M \) as well as certain pairwise distance measurements, the total cost defined by (8) and (6) monotonically decreases during the iterations of Algorithm I at all sensors until the minimum cost is achieved.

**Proof:** See Appendix C.
From Theorem 1 and Theorem 2 we can see that Algorithm I globally converges in the sense of the cost it minimizes. As both theorems do not mention either the convex hull constraint or measurement error, this implies that Algorithm I always globally converges even without the convex hull constraint and/or with measurement errors.

For the case with the convex hull constraint but without measurement error, we have the following theorem.

**Theorem 3.** Given a network containing sensors with unknown positions and anchors with known positions as well as certain pairwise distance measurements. If the network satisfies the convex hull constraint and there is no measurement error, the converged localization solution by Algorithm I via minimizing the total cost defined by (8) and (6) is unique, i.e. the correct positions of all sensors.

**Proof:** See Appendix D.

According to this theorem and Theorem 2, we can see that if the network satisfies the convex hull constraint and there is no measurement error, Algorithm I must globally converge to the correct positions of all sensors.

For the case with the convex hull constraint and measurement errors but small, the correct positions of all sensors, the LS solution, and the converged solution by Algorithm I must be close to each other. Because if the measurement errors decrease to zeros, these three would be the same.

2) ‘Pulled-or-Pushed’ Algorithm – Algorithm II: At a sensor, in each iteration of this iterative self-positioning algorithm, the current estimate on the position of this sensor is updated according to its neighbors to produce same number of new estimates. Different from the ‘pulled-only’ updating of Algorithm I, here the current estimate on the position of the sensor is either pulled or pushed by its every neighbor, to a position such that the newly calculated distance equals to the measured distance. Similar to (4) and (3), we can have the updated estimate on the position of the sensor as

\[
\hat{\mathbf{p}}_i(n+1) = \frac{1}{M_i} \sum_{m \in S_i} \hat{\mathbf{p}}_i^{(m)}(n+1)
\]

(9)

where

\[
\hat{\mathbf{p}}_i^{(m)}(n+1) = \hat{\mathbf{p}}_i(n) + \frac{\mathbf{q}_m(n) - \hat{\mathbf{p}}_i(n)}{\|\mathbf{q}_m(n) - \hat{\mathbf{p}}_i(n)\|} (\|\mathbf{q}_m(n) - \hat{\mathbf{p}}_i(n)\| - r_{im})
\]

\[
= \mathbf{q}_m(n) + \frac{\hat{\mathbf{p}}_i(n) - \mathbf{q}_m(n)}{\|\hat{\mathbf{p}}_i(n) - \mathbf{q}_m(n)\|} r_{im}, \quad m \in S_i.
\]

(10)

Again, we can see that the updating operation at each sensor in each iteration is very much simple.

Below we analyze the convergence property as well as the converged estimation performance of the above iterative self-positioning algorithm when iteratively running at all sensors of the considered network.
Similar to the analysis on Algorithm I, we can have the following analysis on Algorithm II. At any sensor, say sensor $i$, in any iteration, say the one at time $n$, the updating of Algorithm II defined by (9) and (10) can be regarded as the following one-step gradient search in minimizing the LS cost function at a constant step size

$$\hat{p}_i(n+1) = \hat{p}_i(n) - \mu \frac{\partial}{\partial p_i} \sum_{m \in S_i} (\|p_i - q_m\| - r_{im})^2 |_{p_i=\hat{p}_i(n), q_m=q_m(n), m \in S_i}. \quad (11)$$

Further, the updating at any sensor in any iteration also can be regarded as the following one-step gradient search in minimizing the corresponding total LS cost function at step size $\mu = \frac{1}{2M_i}$

$$\hat{p}_i(n+1) = \hat{p}_i(n) - \mu \frac{\partial h_{LS}(p_m, m = 1, ..., M_u)}{\partial p_i} |_{p_i=\hat{p}_i(n), p_m=\hat{p}_m(n), m \in S_i}$$

where

$$h_{LS}(p_m, m = 1, ..., M_u) = \sum_{i=1}^{M_u} \sum_{k=M_u+1}^{M} v_{ik}(\|p_i - p_k\| - r_{ik})^2 + \sum_{i,j=1, i<j}^{M_u} v_{ij}(\|p_i - p_j\| - r_{ij})^2. \quad (13)$$

We have the following theorem for the monotonically decreasing of the total LS cost during the proposed gradient search in Algorithm II.

**Theorem 4.** Given a network containing sensors with unknown positions $p_i, i = 1, ..., M_u$ and anchors with known positions $p_m, m = M_u+1, ..., M$ as well as certain pairwise distance measurements, the total LS cost monotonically decreases during the iterations of Algorithm II at all sensors until the minimum cost is achieved.

**Proof:** See Appendix E.

We also have the following theorem for the local convexity of this total LS cost function.

**Theorem 5.** Given a network containing sensors with unknown positions $p_m, m = 1, ..., M_u$ and anchors with known positions $p_m, m = M_u+1, ..., M$ as well as certain pairwise distance measurements. In the absence of measurement error, the total LS cost function is locally convex w.r.t. the positions of all sensors around these correct positions; in the presence of small measurement errors the total LS cost function is locally convex w.r.t. the positions of all sensors around the LS solution.

**Proof:** See Appendix F.

From Theorem 4 and Theorem 5 we can see that as soon as the current estimates of the positions of all sensors enter into the correctly converging area of Algorithm II, Algorithm II converges to the LS solution.

In fact, the correctly converging area of Algorithm II is far wider than the above mentioned small vicinity of the LS solution. This is more so when the measurement errors are small and/or the sensors are well inside the convex hulls of their neighbors respectively.
3) Combined Algorithm – Algorithm III: We have seen that when iteratively running at all sensors of the considered network, Algorithm I leads to global convergence to the correct positions of all sensors in the absence of measurement error, but might not be optimum if there exist measurement errors; whereas Algorithm II suffers from local convergence, but once correctly converged the converged solution would be the LS solution. We would like to ensure global convergence to the LS solution, so we need to propose a combined algorithm that can efficiently switch from the iteration of Algorithm I to the iteration of Algorithm II, independently at individual sensors only based on locally collected information, to take their advantages. To this end, i) the measurement errors must be sufficiently small such that the converged solution by Algorithm I is well inside the correctly converging area of Algorithm II; ii) a suitable indicator that can indicate the status ‘the current estimate is inside the correctly converging area of Algorithm II’ and can be calculated at individual sensors only based on locally collected information needs to be proposed.

We know that practically the measurement errors are small as compared to their corresponding distances measured. When the measurement errors are small, the converged solution by Algorithm I would be close to the correct positions of all sensors thus the LS solution, because three of them are close to each other for small measurement errors. As mentioned earlier, Sufficient small measurement errors are requested to ensure the converged solution by Algorithm I is well inside the correctly converging area of Algorithm II. Here how small measurement errors can be regarded as ‘sufficiently small’ depends on the geometric placements of the anchors and sensors. If a sensor is well inside the convex hull of its neighbors, not close to the boundary of the convex hull, this requirement is mild.

Generally it is difficult to propose the above mentioned indicator. Here we only propose a pseudo-indicator that can be calculated at individual sensors only based on locally collected information. Our proposed pseudo-indicator is

\[
\eta_i = \max_{m \in S_i} \frac{||\hat{p}_i(n) - q_m(n)|| - r_{im}}{r_{im}}.
\]

For such a ratio, we can set a suitable threshold \(\eta\). If the ratio calculated at an individual sensor is below this threshold, we then approximately regard the current estimate on the position of this sensor has been inside the correctly converging area of Algorithm II, the combined algorithm then switches to the iteration of Algorithm II. Otherwise, the combined algorithm switches back to the iteration of Algorithm I. This ‘switch back’ option improves the robustness of the implementation, resulting smooth converging.

C. Initialization Techniques Improving Convergence Speed

Although Algorithm I is globally convergent in the sense of the cost it minimizes thus can be initialized randomly, in order to improve the resultant convergence speed, it would be better to adopt a suitable initialization. We propose the following initialization at each individual sensor:
• If the sensor has three or more neighbor anchors, the initial estimate on the position of this sensor can be calculated by one of the existing algorithms for range based positioning, such as the one proposed in [26] or [27].
• If the sensor has two neighbor anchors, the initial estimate can be set as their average (i.e. the middle point) of the two possible solutions.
• If the sensor has one neighbor anchor, the initial estimate can be set as the position of this anchor (in practical application of our proposed algorithm a small random bias is needed to avoid division by zero).
• If the sensor has no neighbor anchor, the initial estimate can be set as the average of the initial or updated estimates on the positions of its neighbor sensors (excluding the neighbor sensor that has not yet initial estimate).

D. Relevant Issues

1) Convergence Speed: Among our proposed algorithms, Algorithm I has the ability to converge globally, and the combined algorithm relies on such global convergence. The convergence speed also depends on many aspects. For example, it depends on how many hops sensors have to go to reach an anchor, the closer the sensor, the faster the convergence. However, as the iteration used is non-linear, the converging speed can not be derived in a closed form. We can only show the resultant convergence speed numerically by simulations.

2) Violation of Convex Hull Constraint: From the analysis we can see that Algorithm I always converges globally in the sense of the globally convex cost it minimizes, even without the convex hull constraint and/or with severe measurement errors. If a sensor does not satisfy the convex hull constraint, such convergence would not lead to a localization solution close to its correct position, even in the absence of measurement error. But this does not affect the optimum convergence of other sensors with the convex hull constraint as long as their neighbors are localizable.

IV. Simulation Study

In this section we evaluate by simulations the localization performance of our proposed iterative self-positioning algorithms when iteratively running at all sensors of the considered network. We design three simulation settings as shown in Fig.1, each for different purpose. For all three of them, we assume the measurement errors as i.i.d. zero-mean Gaussian noises. In addition, we assume that the iterations running at all sensors in parallel and independently are fast enough as compared to the corresponding updating session, to avoid interferences from each other. Here the updating session refers to the time period for updating during which the received data from neighbors remain unchanged.
The first simulation setting is shown in Fig.1(a), where there are two sensors to be localized and four anchors at the four corners. This simulation setting is designed to show the convergence property and converged estimation performance of our proposed algorithms and their relationship with the measurement error strength.

Fig.2(a) shows the r.a.m.s. positioning errors of Algorithm I, Algorithm II initialized from the correct positions (just for comparison), the combined algorithm with different thresholds, as well as the square root of the CRLB, against the average iterations passed at every sensor. Here the common r.m.s. measurement error in the inter-neighbor distance measurements is set as $\sqrt{10}$ length units, and the curves are obtained by averaging over 1,000 independent simulation runs. From the figure we can see that Algorithm I always converges globally, but its converged estimation performance has a gap as compared to the CRLB; whereas Algorithm II with the correct initialization achieves the CRLB; with a threshold $1/2$ or lower the combined algorithm also converges and achieves the CRLB. All algorithms converge at a fast convergence speed (around 20 average iterations passed at every sensor).

Fig.2(b) shows the converged r.a.m.s. positioning errors of these algorithms against the r.m.s. measurement error. The curves are obtained by averaging over 1,000 independent simulation runs. From the figure we can see that within a large range of r.m.s. measurement errors all our proposed algorithms work well. All these algorithms except Algorithm I converge and achieve the CRLB. Algorithm I always converges globally, but its converged estimation performance has a gap as compared with the CRLB. We notice that for large measurement errors, the converged estimation performance of Algorithm I might be better than the CRLB. The reason is as follows: 1. The converged localization solution of Algorithm I is biased, because the cost function Algorithm I adopts came directly from the least square cost (which leads to an unbiased solution as in Algorithm II) by simply excluding the contribution of partial measurements (which results in ‘pulled-only’ updating). 2. Algorithm I tends to converge to a solution within the convex hull, which is consistent to the convex hull constraint. 3. Algorithm I may have more solutions with the same minimum cost, i.e. 0. This results in that as soon as the updated estimate enters into the 0-cost area, the estimate might stay there unchanged. So the converged estimation performance depends on the searching route, and for some searching routes the converged solutions might have better estimation performance than the CRLB.

Fig.2(c) and Fig.2(d) show the simulation results of Fig.2(a) and Fig.2(b) when all algorithms except Algorithm II are initialized using the proposed initialization. We can see that the convergence speed has been significantly improved. We also notice that due to the introduced initialization, the converged estimation performance of Algorithm I is slightly different from (here better than) that obtained by random initialization because of the different searching routes.
The second simulation setting is shown in Fig.1(b), where there are six sensors to be localized and three anchors at three corners. This simulation setting is designed for algorithm comparison. In the algorithm comparison we include the algorithm in [8] without LS refinement as a typical distributed cooperative iterative algorithm. We also include the centralized iterative extended MDS algorithm with LS refinement.

Fig.3 shows the r.a.m.s. positioning errors of the algorithms, all adopting our proposed initialization, against the average iterations passed at every sensor. Here the common r.m.s. measurement error is set as $\sqrt{10}$ length units, and the curves are obtained by averaging over 1,000 independent simulation runs. From the figure we can see the following: All our proposed algorithms converge faster than the algorithm in [8] without LS refinement, and all our proposed algorithms except Algorithm I achieve the CRLB. Algorithm I converges quickly due to the proposed initialization used, but then randomly drifts to a lower estimation performance level due to the cost this algorithm minimizes. The centralized iterative extended MDS algorithm with LS refinement can achieve the CRLB, but as mentioned earlier the centralized algorithm need all information to be collected and transferred to a processing center, resulting very high computational complexity.

The third simulation setting is shown in Fig.1(c), where a relatively large wireless sensor network is considered. Here ‘large’ refers to not only the number of the sensors in the network, but also the number of hops sensors have to go to reach an anchor. In this network, there are 121 nodes roughly located on a regular grid, for which the four nodes on the four corners are position-known anchors and all others are sensors to be localized. The cooperative neighborhood is as small as that only containing the immediate neighbors on grid. We notice that the sensor at the middle of the network has to go 10 hops to reach an anchor.

Fig.4 shows the r.a.m.s. positioning errors of our proposed algorithms against the average iterations passed at every sensor, all adopting our proposed initialization. Here the common r.m.s. measurement error is set as 1 length unit, and the curves are obtained by averaging over 100 independent simulation runs. From the figure we can see that within around 200 iterations all algorithms converge: Algorithm I converges to a level that is slightly different from the CRLB (here it is better than the CRLB due to the proposed initialization and resultant searching routes); Algorithm II with the correct initialization converges to the CRLB; the combined algorithm also converges to the CRLB. In the simulations we also notice that when the number of hops sensors have to go to reach an anchor is large, as the case in this simulation setting, the convergence speed of the localization process might be slow if randomly initialized. This is due to the wave-like disturbances caused by the asynchronous updating of the iterative self-positioning algorithm used.
V. CONCLUSION

In this paper we have proposed a distributed cooperative localization scheme and several iterative self-positioning algorithms. Analysis and simulation show that when iteratively running at all sensors of the considered network, Algorithm I ensures global convergence to the correct positions of all sensors in the absence of measurement error, or their close vicinity if there exist measurement errors but small; Algorithm II suffers from local convergence, but once correctly converged the converged solution would be the LS solution; the third algorithm, i.e. the combined algorithm that switches between the iterations of Algorithm I and Algorithm II efficiently and independently at individual sensors only based on locally collected information, ensures global convergence to the LS solution as long as the measurement errors are sufficiently small such that the converged solution by Algorithm I is well inside the correctly converging area of Algorithm II.

APPENDIX

A. Average CRLB on Average Mean Squared Positioning Error

The average CRLB is [2]

\[
\text{average CRLB} = \frac{1}{M_u} \text{trace}\{I^{-1}\} \tag{15}
\]

where \( \text{trace} \) is the trace operator, and \( I \), the Fisher Information Matrix (FIM), is

\[
I = -E \left\{ \left( \begin{array}{ccc}
\frac{\partial^2 \ln f}{\partial p_i \partial p_j^T} & \cdots & \frac{\partial^2 \ln f}{\partial p_i \partial p_{M_u}^T} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 \ln f}{\partial p_{M_u} \partial p_{M_u}^T} & \cdots & \frac{\partial^2 \ln f}{\partial p_{M_u} \partial p_{M_u}^T}
\end{array} \right) \right\}. \tag{16}
\]

Here

\[
f(r_{ij}, i, j \in \{i < j, v_{ij} = 1\}|p_m, m = 1, \ldots, M) = \frac{1}{\sqrt{(2\pi\sigma_n^2)^M \sum_{i<j,i,j=1}^M v_{ij}(\|p_i-p_j\|-r_{ij})^2}} e^{-\frac{1}{2\sigma_n^2}(\sum_{i<j,i,j=1}^M v_{ij}(\|p_i-p_j\|-r_{ij})^2)} \tag{17}
\]

where \( v_{ij} = v_{ji} \) is 1 if node \( i \) and node \( j \) are neighbors, and 0 otherwise. For \( i = j = 1, \ldots, M_u \)

\[
-E \left\{ \frac{\partial^2 \ln f}{\partial p_i \partial p_i^T} \right\} = \frac{1}{\sigma_n^2} \sum_{m \neq i,m=1}^{M_u} v_{im} \frac{(p_i - p_m)(p_i - p_m)^T}{\|p_i - p_m\|^2} \tag{18}
\]

and for \( i, j = 1, \ldots, M_u \)

\[
-E \left\{ \frac{\partial^2 \ln f}{\partial p_i \partial p_j^T} \right\} = -\frac{1}{\sigma_n^2} v_{ij} \frac{(p_i - p_j)(p_i - p_j)^T}{\|p_i - p_j\|^2}. \tag{19}
\]
B. Proof of Theorem 1

Proof: For the total cost function defined by (8) and (6), its Hessian matrix w.r.t. the composed vector of the positions of all sensors is

\[
I_h = \nabla^2 h = \sum_{i=1}^{M_u} \sum_{k=M_u+1}^{M} v_{ik} \nabla g(p_i, p_k) + \sum_{i<j, i,j=1}^{M_u} v_{ij} \nabla g(p_i, p_j)
\]

(20)

where

\[
\nabla = \begin{pmatrix}
\frac{\partial^2}{\partial p_1 \partial p_1} & \cdots & \frac{\partial^2}{\partial p_1 \partial p_{M_u}} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2}{\partial p_{M_u} \partial p_1} & \cdots & \frac{\partial^2}{\partial p_{M_u} \partial p_{M_u}}
\end{pmatrix}.
\]

(21)

For \(i = 1, \ldots, M_u\) and \(k = M_u + 1, \ldots, M\), \(g(p_i, p_k)\) is a convex function w.r.t. \(p_i\) if \(p_k\) is fixed, its Hessian matrix w.r.t. \(p_i\) must be positive semidefinite, so does its Hessian matrix w.r.t. the composed vector \([p_i^T, \ldots, p_{M_u}^T]^T\). Similarly, for \(i, j = 1, \ldots, M_u\) and \(i < j\), \(g(p_i, p_j)\) is a convex function w.r.t. \([p_i^T, p_j^T]^T\), its Hessian matrix w.r.t. \([p_i^T, p_j^T]^T\) must be positive semidefinite, so does its Hessian matrix w.r.t. \([p_i^T, \ldots, p_{M_u}^T]^T\). Therefore, the Hessian matrix of the total cost function \(h\) w.r.t. \([p_1^T, \ldots, p_{M_u}^T]^T\), i.e. the sum of several/limited positive semidefinite Hessian matrices, is also a positive semidefinite matrix. Thus the total cost function is a (globally) convex function w.r.t. \([p_1^T, \ldots, p_{M_u}^T]^T\), the composed vector of the positions of all sensors.

This proves the theorem.

\[\blacksquare\]

C. Proof of Theorem 2

Proof: We firstly consider the change of the total costs before and after the updating at any sensor, say sensor \(i\), in any iteration, say the one at time \(n\), and assume that from time \(n\) to time \(n+1\) only this updating has been conducted throughout the whole network. Such change can be represented as

\[
h(\hat{P}_1(n+1), \hat{P}_2(n+1), \ldots, \hat{P}_{M_u}(n+1)) - h(\hat{P}_1(n), \hat{P}_2(n), \ldots, \hat{P}_{M_u}(n))
\]

\[
= h(\hat{P}_1(n), \hat{P}_2(n+1), \ldots, \hat{P}_{M_u}(n)) - h(\hat{P}_1(n), \hat{P}_2(n), \ldots, \hat{P}_{M_u}(n))
\]

\[
= \sum_{m \in S_{ic}(n+1)} \left(\|\hat{P}_i(n+1) - q_{im}(n)\| - r_{im}\right)^2 - \sum_{m \in S_{ic}(n)} \left(\|\hat{P}_i(n) - q_{im}(n)\| - r_{im}\right)^2,
\]

(22)

where \(S_{ic}(n)\) is the set containing sensor \(i\)'s neighbors that did pull in the updating at time \(n\).

An illustrative explanation can be shown in Fig.5, where without loss of generality there exist two neighbors, one keeps and one pulls the current estimate of the position of the sensor. The positions of the corresponding points are as follows. A: \(\hat{P}_i(n)\), also \(\hat{P}_i^{(1)}(n+1)\); B1: \(q_1(n)\); B2: \(q_2(n)\); C2: \(\hat{P}_i^{(2)}(n+1)\), also \(\hat{P}_i^{(2)}(n+2)\); D: \(\hat{P}_i(n+1)\), average of A and C2; E1: \(\hat{P}_i^{(1)}(n+2)\); F: \(\hat{P}_i(n+2)\), average of E1 and C2.


We have the following relationship

\[
\sum_{m \in S_{i}\,(n+1)} (\|\hat{\mathbf{p}}_i(n+1) - \mathbf{q}_m\| - r_{im})^2 = \sum_{m \in S_{i}\,(n+1)} \|\hat{\mathbf{p}}_i(n+1) - \hat{\mathbf{p}}^{(m)}_i(n+2)\|^2 \\
\leq \sum_{m \in S_{i}\,(n+1)} \|\hat{\mathbf{p}}_i(n+1) - \hat{\mathbf{p}}^{(m)}_i(n+1)\|^2 \tag{23}
\leq \sum_{m \in S_i} \|\hat{\mathbf{p}}_i(n) - \hat{\mathbf{p}}^{(m)}_i(n+1)\|^2 \tag{24}
\leq \sum_{m \in S_i} \|\hat{\mathbf{p}}_i(n) - \hat{\mathbf{p}}^{(m)}_i(n+1)\|^2 \tag{25}
= \sum_{m \in S_i(\,n)} \|\hat{\mathbf{p}}_i(n) - \hat{\mathbf{p}}^{(m)}_i(n+1)\|^2 \\
= \sum_{m \in S_i(\,n)} (\|\hat{\mathbf{p}}_i(n) - \mathbf{q}_m\| - r_{im})^2. \tag{26}
\]

We can see the following: (24) holds because for each \( m \in S_{i}(n+1) \) we have

\[
\|\hat{\mathbf{p}}_i(n+1) - \hat{\mathbf{p}}^{(m)}_i(n+2)\| \leq \|\hat{\mathbf{p}}_i(n+1) - \hat{\mathbf{p}}^{(m)}_i(n+1)\|. \tag{27}
\]

Here we utilized the fact that the shortest distance from a point outside a circle to this circle is the distance on the direction from this point to the center of the circle. (25) holds because the more positive values are added. (26) holds because the average of several/limited points has the minimum sum of squared distances to these points. All equations hold if and only if \( \hat{\mathbf{p}}_i(n+1) = \hat{\mathbf{p}}^m_i(n) \), which means that the corresponding cost has achieved its minimum. This demonstrates that the total cost according to which Algorithm I gradient-searches decreases during the updating at each sensor in each iteration, and remains the same if the minimum cost is achieved.

The whole gradient search at all sensors is composed of the same updating running in parallel and independently, so the total cost monotonically decreases until the minimum cost is achieved.

This proves the theorem.

D. Proof of Theorem 3

\textit{Proof:} If the network satisfies the convex hull constraint and there is no measurement error, the total cost defined by equations (8) and (6) only becomes zero at the correct positions of all sensors. Here zero is also the minimum total cost, because the total cost defined by equations (8) and (6) can not be negative. According to Theorem 1 the total cost function is (globally) convex w.r.t. the positions of all sensors, which means that globally there is only one localization solution with the minimum total cost, i.e. the correct positions of all sensors with zero cost.

This proves the theorem.
E. Proof of Theorem 4

\textbf{Proof:} We firstly consider the change of the total costs before and after the updating at any sensor, say sensor $i$, in any iteration, say the one at time $n$, and assume that from time $n$ to time $n+1$ only this updating has been conducted throughout the whole network. Such change can be represented as

$$h_{LS} \left[ \hat{p}_1(n+1), ... \hat{p}_i(n+1), ... \hat{p}_{M_u}(n+1) \right] - h_{LS} \left[ \hat{p}_1(n), ... \hat{p}_i(n), ... \hat{p}_{M_u}(n) \right]$$

$$= h_{LS} \left[ \hat{p}_1(n), ... \hat{p}_i(n+1), ... \hat{p}_{M_u}(n) \right] - h_{LS} \left[ \hat{p}_1(n), ... \hat{p}_i(n), ... \hat{p}_{M_u}(n) \right]$$

$$= \sum_{m \in S_i} \left[ \| \hat{p}_i(n+1) - q_m \| - r_{im} \right]^2 - \sum_{m \in S_i} \left[ \| \hat{p}_i(n) - q_m \| - r_{im} \right]^2,$$

(28)

An illustrative explanation can be shown in Fig.6 where without loss of generality there exist two neighbors, one pushes and one pulls the current estimate of the position of the sensor in the considered updating. The positions of the corresponding points are as follows: A: $\hat{p}_i(n)$; B1: $q_1(n)$; B2: $q_2(n)$; C1: $\hat{p}_i^{(1)}(n+1)$; C2: $\hat{p}_i^{(2)}(n+1)$; D: $\hat{p}_i(n+1)$, average of points C1 and C2; E1: $\hat{p}_i^{(1)}(n+2)$; E2: $\hat{p}_i^{(2)}(n+2)$; F: $\hat{p}_i(n+2)$, average of E1 and E2.

We have the following relationship

$$\sum_{m \in S_i} \left[ \| \hat{p}_i(n+1) - q_m \| - r_{im} \right]^2 = \sum_{m \in S_i} \| \hat{p}_i(n+1) - \hat{p}_i^{(m)}(n+2) \|^2$$

$$\leq \sum_{m \in S_i} \| \hat{p}_i(n+1) - \hat{p}_i^{(m)}(n+1) \|^2$$

(29)

$$\leq \sum_{m \in S_i} \| \hat{p}_i(n) - \hat{p}_i^{(m)}(n+1) \|^2$$

(30)

$$= \sum_{m \in S_i} \left( \| \hat{p}_i(n) - q_m \| - r_{im} \right)^2.$$  

(31)

We can see the following: (30) holds because for $m \in S_i$ we have

$$\| \hat{p}_i(n+1) - \hat{p}_i^{(m)}(n+2) \| \leq \| \hat{p}_i(n+1) - \hat{p}_i^{(m)}(n+1) \|.$$  

(32)

Here we utilized the fact that the distance of a point off a circle to this circle is the shortest on the direction from/towards the circle center. (31) holds because the average of several/limited points has the minimum sum of squared distances to these points. All equations hold if and only if $\hat{p}_i(n+1) = \hat{p}_i(n)$, which means that the total LS cost has achieved its minimum. This demonstrates that the total LS cost according to which Algorithm II searches based on gradient decreases during the updating at each sensor in each iteration, and remains the same if the minimum cost is achieved.

The whole gradient search at all sensors is composed of the same updating running in parallel and independently, so the total cost monotonically decreases until the minimum cost is achieved.

This proves the theorem.
F. Proof of Theorem 5

Proof: For the total LS cost function, its Hessian matrix w.r.t. the positions of all sensors is

\[
\mathbf{I}_{h_{LS}} = \nabla^2 h_{LS} = \sum_{i=1}^{M_u} \sum_{k=M_u+1}^{M} v_{ik} \nabla^2 (\| \mathbf{p}_i - \mathbf{p}_k \| - r_{ik})^2 + \sum_{i,j=1,i<j}^{M_u} v_{ij} \nabla^2 (\| \mathbf{p}_i - \mathbf{p}_j \| - r_{ij})^2.
\]

In the absence of measurement error, \((\| \mathbf{p}_i - \mathbf{p}_k \| - r_{ik})^2\) for \(i = 1, \ldots, M_u\) and \(k = M_u+1, \ldots, M\) is locally convex w.r.t. \(\mathbf{p}_i\) around \(\mathbf{p}_i\), thus its Hessian matrix w.r.t. \(\mathbf{p}_i\) around \(\mathbf{p}_i\) must be positive semidefinite, so does its Hessian matrix w.r.t. \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\) around \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\); Similarly, \((\| \mathbf{p}_i - \mathbf{p}_j \| - r_{ij})^2\) is locally convex w.r.t. \([\mathbf{p}_i^T, \mathbf{p}_j^T]^T\) around \([\mathbf{p}_i^T, \mathbf{p}_j^T]^T\), thus its Hessian matrix w.r.t. \([\mathbf{p}_i^T, \mathbf{p}_j^T]^T\) around \([\mathbf{p}_i^T, \mathbf{p}_j^T]^T\) is positive semidefinite, so does its Hessian matrix w.r.t. \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\) around \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\). Therefore, the Hessian matrix of the total LS cost function w.r.t. \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\) around \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\), the sum of several positive semidefinite Hessian matrices, is also a positive semidefinite matrix. Thus the total LS cost function is a locally convex function w.r.t. \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\) around \([\mathbf{p}_1^T, \ldots, \mathbf{p}_{M_u}^T]^T\).

In the presence of small measurement errors, the above derivations hold approximately, thus the local convexity of all items in the total LS cost function holds around the LS solution, so does the local convexity of the total LS cost function.

This proves the theorem.

\[\square\]

REFERENCES


Fig. 1. The three localizable networks: (a) a simple network with two sensors to be localized; (b) another simple network; (c) a relatively large network: sensors to be localized ('+'), position-known anchors ('o'), available inter-neighbor connections and measurements ('· · ·').
Fig. 2. The estimation performance of our proposed algorithms: (a) the r.a.m.s. positioning error versus the average number of iterations passed at every sensor, with r.m.s. measurement error $\sqrt{\Omega}$. (b) the converged r.a.m.s. positioning error versus the r.m.s. measurement error. (c)(d) the simulation results of (a)(b) when adopting our proposed initialization. In all simulations Algorithm II is initialized from the correct positions of all sensors. All curves are obtained by averaging over 1,000 independent simulation runs.
Fig. 3. The r.a.m.s. positioning error versus the average number of iterations passed at every sensor. The r.m.s. measurement error is $\sqrt{10}$. All curves are obtained by averaging over 1,000 independent simulation runs.

Fig. 4. The r.a.m.s. positioning error versus the average number of iterations passed at every sensor. The r.m.s. measurement error is 1. All curves are obtained by averaging over 100 independent simulation runs.
Fig. 5. An illustration for explaining the iteration of Algorithm I at sensor $i$ at time $n$, where there exist two neighbors. A: $\hat{p}_i(n)$, also $\hat{p}_i^{(1)}(n+1)$; B1: $q_1$; B2: $q_2$; C2: $\hat{p}_i^{(2)}(n+1)$, also $\hat{p}_i^{(2)}(n+2)$; D: $\hat{p}_i(n+1)$, average of A and C2; E1: $\hat{p}_i^{(1)}(n+2)$; F: $\hat{p}_i(n+2)$, average of E1 and C2.

Fig. 6. An illustration for explaining the iteration of Algorithm II at sensor $i$ at time $n$, where there exist two neighbors. A: $\hat{p}_i(n)$; B1: $q_1$; B2: $q_2$; C1: $\hat{p}_i^{(1)}(n+1)$; C2: $\hat{p}_i^{(2)}(n+1)$; D: $\hat{p}_i(n+1)$, average of points C1 and C2; E1: $\hat{p}_i^{(1)}(n+2)$; E2: $\hat{p}_i^{(2)}(n+2)$; F: $\hat{p}_i(n+2)$, average of E1 and E2.