Distributed Sensor Localization in Random Environments Using Minimal Number of Anchor Nodes

Usman A. Khan, *Student Member, IEEE*, Soummya Kar, *Student Member, IEEE*, and José M. F. Moura, *Fellow, IEEE*

***Abstract*—The paper introduces DILOC, a *distributed, iterative* algorithm to locate M sensors (with unknown locations) in $\mathbb{R}^m$, $m \geq 1$, with respect to a minimal number of $m+1$ anchors with known locations. The sensors and anchors, nodes in the network, exchange data with their neighbors only; no centralized data processing or communication occurs, nor is there a centralized fusion center to compute the sensors' locations. DILOC uses the barycentric coordinates of a node with respect to its neighbors; these coordinates are computed using the Cayley–Menger determinants, i.e., the determinants of matrices of internode distances. We show convergence of DILOC by associating with it an absorbing Markov chain whose absorbing states are the states of the anchors. We introduce a stochastic approximation version extending DILOC to random environments, i.e., when the communications among nodes is noisy, the communication links among neighbors may fail at random times, and the internodes distances are subject to errors. We show a.s. convergence of the modified DILOC and characterize the error between the true values of the sensors' locations and their final estimates given by DILOC. Numerical studies illustrate DILOC under a variety of deterministic and random operating conditions.**

***Index Terms*—Absorbing Markov chain, anchor, barycentric coordinates, Cayley–Menger determinant, distributed iterative sensor localization, sensor networks, stochastic approximation.**

I. Introduction

LOCALIZATION is a fundamental problem in sensor networks. Information about the location of the sensors is key to process the sensors' measurements accurately. In applications where sensors are deployed randomly, they have no knowledge of their exact locations, but equipping each of them with a localization device like a GPS is expensive, not robust to jamming in military applications, and is usually of limited use in indoor environments. Our goal is to develop a distributed (decentralized) localization algorithm where the sensors find their locations under a limited set of assumptions and conditions. In this paper, we assume that there are $m+1$ *anchors*, i.e., nodes in the network that know their coordinates in $\mathbb{R}^m$ space; for example, $m=2$ corresponds to nodes lying on a plane, while for $m=3$ the nodes are in three dimensional Euclidean space. The goal is then to locate the remaining M nodes in the network, which we call *sensors* and which do not know their coordinates, with respect to the anchors.[1] The problem is practically interesting when $M \gg m+1$. Two important characteristics in our work are i) we assume that the sensors lie in the convex hull of the anchors; and, from this, it follows that ii) each sensor can find $m+1$ nodes (i.e., a possible combination of anchors and sensors) such that it lies in the convex hull of these $m+1$ nodes. The paper will discuss the practical significance of these assumptions.

With networks of interest in applications, the distance of most of the M sensors to the $m+1$ anchors is large, so that it is impractical for the sensors to communicate *directly* with the anchors. Further, because M is assumed very large, to compute the locations of the sensors at a central station is not feasible, as it would require a large communication effort, expensive large-scale computation, and add latency and bottlenecks to the network operation. These networks call for efficient *distributed* algorithms where each node communicates directly only with a few neighboring nodes (either sensors or anchors) and a low order computation is performed locally at the node and at each iteration of the algorithm, for example, see [1]. We present here the Distributed Iterative LOCalization algorithm (DILOC, pronounced *die-lock*) that overcomes the above challenges in large-scale randomly deployed networks.

In DILOC, the sensors start with an initial estimate of their locations, for example, a random guess. This random guess is arbitrary and does not need to place the sensors in the convex hull of the anchors. The sensors then update their locations, which we call the state of the network, by exchanging their state information *only* with a carefully chosen subset of $m+1$ of their neighbors [see ii) above]. This state updating is a convex combination of the states of the neighboring nodes. The coefficients of the convex combination are the barycentric coordinates of the sensors with respect to their neighbors [2], [3], which are determined from the Cayley–Menger determinants. These are the determinants of matrices that collect the local internode distances,

Manuscript received February 24, 2008; accepted December 30, 2008. First published February 06, 2009; current version published April 15, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pramod K. Varshney. All authors contributed equally to the paper. This work was supported in part by the DARPA DSO Advanced Computing and Mathematics Program Integrated Sensing and Processing (ISP) Initiative under ARO Grant DAAD 19-02-1-0180, by NSF under Grants ECS-0225449 and CNS-0428404, by ONR under Grant MURI-N000140710747, and by an IBM Faculty Award.

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: ukhan@ece.cmu.edu; soummyak@andrew.cmu.edu; moura@ece.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2009.2014812

[1] In the sequel, the term *node* refers to either *anchors* (known locations) or *sensors* (locations to be determined). In a few exceptions, easily resolved from the context, we will still write *sensors*, when we actually mean nodes.

i.e., the distances among the sensors or among the sensors and the anchors.

DILOC is distributed and iterative; each node updates locally its own state and then sends its state information to its neighbors; nowhere does DILOC need a fusion center or global communication. We prove almost sure (a.s.) convergence of DILOC in both deterministic and random network environments by showing that DILOC behaves as an absorbing Markov chain, where the anchors are the absorbing states. We prove convergence under a broad characterization of noise. In particular, we consider three types of randomness, acting simultaneously. These model many practical random sensing and communication distortions as, for example, when: i) the internode distances are known up to random errors, which is common in cluttered environments and also in ad hoc environments, where cheap low resolution sensors are deployed; ii) the communication links between the nodes fail at random times; this is mainly motivated by wireless digital communication, where packets may get dropped randomly at each iteration, particularly, if the nodes are power limited or there are bandwidth or communication rate constraints in the network; and iii) the communication among two nodes, when their communication link is active, is corrupted by noise.

Although a node can only communicate directly with its neighbors (e.g., nodes within a small radius), we assume that, when the links are deterministic and never fail, the network graph is connected, i.e., there is a communication path (by multihop) between any arbitrary pair of nodes. In a random environment, internode communication links may not stay active all the time and are subject to random failures. Consequently, there may be iterations when the network is not connected; actually, there might never be iterations when the network is connected. We will show under broad conditions almost sure convergence of an extended version of DILOC that we term the Distributed Localization in Random Environments (DLRE) algorithm. DLRE employs stochastic approximation techniques using a decreasing weight sequence in the iterations.

In the following, we contrast our work with the existing literature on sensor localization.

Brief Review of the Literature: The literature on localization algorithms may be broadly characterized into centralized and distributed algorithms. Illustrative centralized localization algorithms include: maximum likelihood estimators that are formulated when the data is known to be described by a statistical model [4], [5]; multidimensional scaling (MDS) algorithms that formulate the localization problem as a least squares problem at a centralized location [6], [7]; work that exploits the geometry of the Euclidean space, like when locating a single robot using trilateration in $m = 3$-dimensional space (see [8]), where a geometric interpretation is given to the traditional algebraic distance constraint equations; localization algorithms with imprecise distance information (see [9]), where the authors exploit the geometric relations among the distances in the optimization procedure; for additional work, see, e.g., [10] and [11]. Centralized algorithms are fine in small or tethered network environments; but in large untethered networks, they incur high communication cost and may not be scalable; they depend on the availability and robustness of a central processor and have a single point of failure.

Distributed localization algorithms can be characterized into two classes: multilateration and successive refinements. In multilateration algorithms [12]–[15], each sensor estimates its range from the anchors and then calculates its location via multilateration [16]. The multilateration scheme requires a high density of anchors, which is a practical limitation in large sensor networks. Further, the location estimates obtained from multilateration schemes are subject to large errors because the estimated sensor-anchor distance in large networks, where the anchors are far apart, is noisy. To overcome this problem, a high density of anchors is required. We, on the other hand, do not estimate distances to far-away nodes. Only local distances to nearby nodes are estimated; these should have better accuracy. This allows us to employ the minimal number $m + 1$ of anchors (for localization in $\mathbb{R}^m$).

A distributed multidimensional scaling algorithm is presented in [17]. Successive refinement algorithms that perform an iterative minimization of a cost function are presented in [18]–[20]. Reference [18] discusses an iterative scheme where they assume 5% of the nodes as anchors. Reference [20] discusses a Self-Positioning Algorithm (SPA) that provides a GPS-free positioning and builds a relative coordinate system.

Another formulation to solve localization problems in a distributed fashion is the probabilistic approach. Nonparametric belief propagation on graphical models is used in [21]. Sequential Monte Carlo methods for mobile localization are considered in [22]. Particle filtering methods have been addressed in [23] where each sensor stores representative particles for its location that are weighted according to their likelihood. Reference [24] tracks and locates mobile robots using such probabilistic methods.

Completion of partially specified distance matrices is considered in [25] and [26]. The approach is relevant when the (entire) partially specified distance matrix is available at a central location. The algorithms complete the unspecified distances under the geometrical constraints of the underlying network. The key point to note in our work is that DILOC is *distributed*. In particular, it does not require a centralized location to perform the computations.

In comparison with these references, DILOC is equivalent to solving by a *distributed* and *iterative* algorithm a large system of linear algebraic equations where the system matrix is highly sparse. Our method exploits the structure of this matrix, which results from the topology of the communication graph of the network. We prove the a.s. convergence of the algorithm under broad noise conditions and characterize the bias and mean square error properties of the estimates of the sensor locations obtained by DILOC.

We divide the rest of the paper into two parts. The first part of the paper is concerned with the deterministic formulation of the localization problem and consists of Sections II–IV. Section II presents preliminaries and then DILOC, the distributed iterative localization algorithm, that is based on barycentric coordinates, generalized volumes, and Cayley–Menger determinants. Section III proves DILOC's convergence. Section IV presents the DILOC-REL, DILOC with relaxation, and proves that it asymptotically reduces to the deterministic case without relaxation. The second part of the paper consists of Sections V–VI and considers distributed localization in random noisy environments. Section V characterizes the random noisy environments and the iterative algorithm for these conditions. Section VI proves the convergence of the distributed localization algorithm in the noisy case that relies on a result on the convergence of Markov processes. Finally, we present detailed numerical simulations in Section VII and conclude the paper in Section VIII. Appendices I–III provide a necessary test, the Cayley–Menger determinant, and background material on absorbing Markov chains.

II. Distributed Sensor Localization: DILOC

In this section, we formally state DILOC in m-dimension Euclidean space, $\mathbb{R}^m (m \geq 1)$, and introduce the relevant notation. Of course, for sensor localization, $m=1$ (sensors in a straight line), $m=2$ (plane), or $m=3$ (3D-space). The generic case of $m>3$ is of interest, for example, when the graph nodes represent m-dimensional feature vectors in classification problems, and the goal is still to find in a distributed fashion their global coordinates (with respect to a reference frame). Since our results are general, we deal with m-dimensional "localization," but, for easier accessibility, the reader may consider $m=2$ or $m=3$. To provide a quantitative assessment on some of the assumptions underlying DILOC, we will, when needed, assume that the deployment of the sensors in a given region follows a Poisson distribution. This random deployment is often assumed and is realistic; we use it to derive probabilistic bounds on the deployment density of the sensors and on the communication radius at each sensor; these can be straightforwardly related to the values of network field parameters (like transmitting power or signal-to-noise ratio) in order to implement DILOC. We discuss the computation/communication complexity of the algorithm and provide a simplistic, yet insightful, example that illustrates DILOC.

A. Notation

Recall that the sensors and anchors are in $\mathbb{R}^m$. Let Θ be the set of nodes in the network decomposed as

$$\Theta = \kappa \cup \Omega \tag{1}$$

where κ is the set of anchors, i.e., the nodes whose locations are known, and Ω is the set of sensors whose locations are to be determined. By $|\cdot|$ we mean the cardinality of the set, and we let $|\Theta| = N, |\kappa| = m+1$, and $|\Omega| = M$. For a set Ψ of nodes, we denote its convex hull by $\mathcal{C}(\Psi)$.[2] For example, if Ψ is a set of three noncollinear nodes in a plane, then $\mathcal{C}(\Psi)$ is a triangle. Let A_Ψ be the generalized volume (area in $m=2$, volume in $m=3$, and their generalization in higher dimensions) of $\mathcal{C}(\Psi)$. Let d_{lk} be the Euclidean distance between two nodes $l,k \in \Theta$, their internode distance; the neighborhood of node l in a given radius, r_l, is

$$\mathcal{K}(l, r_l) = \{k \in \Theta : d_{lk} < r_l\}. \tag{2}$$

Note that $\mathcal{K}(l, r_l)$ may contain anchors as well as sensors. In $\mathcal{K}(l, r_l)$, the subset Θ_l introduced in Lemma 1 and (6) below will play an important role.

Let $\mathbf{c}_l$ be the m-dimensional coordinate vector for node, $l \in \Theta$, with respect to a global coordinate system, written as the m-dimensional row vector

$$\mathbf{c}_l = [c_{l,1}, c_{l,2}, \ldots, c_{l,m}]. \tag{3}$$

The true (possibly unknown) location of node l is represented by $\mathbf{c}_l^*$. Because the distributed localization algorithm DILOC is iterative, $\mathbf{c}_l(t)$ will represent the estimated location vector, or state, for node l at iteration t.

B. Distributed Iterative Localization Algorithm

We state explicitly the assumptions that we make when developing DILOC.

B0) Convexity: All the sensors lie inside the convex hull of the anchors

$$\mathcal{C}(\Omega) \subseteq \mathcal{C}(\kappa). \tag{4}$$

B1) Anchor Nodes: The anchors' locations are known, i.e., their state remains constant

$$\mathbf{c}_q(t) = \mathbf{c}_q^*, \quad q \in \kappa, t \geq 0. \tag{5}$$

B2) Nondegeneracy: The generalized volume for $\kappa, A_\kappa \neq 0$.[3]

From B0, the next Lemma follows easily.

Lemma 1 (Triangulation): For every sensor $l \in \Omega$, there *exists* some $r_l > 0$ such that a *triangulation set*, $\Theta_l(r_l)$, satisfying the following conditions:

$$\begin{aligned} &\Theta_l(r_l) \subseteq \mathcal{K}(l, r_l), \quad l \notin \Theta_l(r_l), l \in \mathcal{C}(\Theta_l(r_l)) \\ &|\Theta_l(r_l)| = m+1, \quad A_{\Theta_l(r_l)} \neq 0 \end{aligned} \tag{6}$$

exists.[4]

Proof: Clearly, by B0, κ satisfies (6) and by taking $r_l = \max_{l,k} d_{lk}, (l \in \Omega, k \in \kappa)$ the Lemma follows. ■

Lemma 1 provides an existence proof, but, in localization in wireless sensor networks, it is important to triangulate a sensor not with the network diameter but with a small r_l. In fact, Section II-D discusses the probability of finding one such Θ_l with $r_l \ll \max_{l,k} d_{lk}, (l \in \Omega, k \in \Theta)$. In addition, Appendix I provides a procedure to test the convex hull inclusion of a sensor, i.e., for any sensor, l, to determine if it lies in the convex hull of $m+1$ nodes arbitrarily chosen from the set, $\mathcal{K}(l, r_l)$, of its neighbors. Finding Θ_l is an important step in DILOC and we refer to it as *triangulation*.

[2] The convex hull, $\mathcal{C}(\Psi)$, of a set of points in Ψ is the smallest convex set containing Ψ.

[3] Nondegeneracy simply states that the anchors do not lie on a hyperplane. If this was the case, then the localization problem reduces to a lower dimensional problem, i.e., $\mathbb{R}^{m-1}$ instead of $\mathbb{R}^m$. For instance, if all the anchors in the network lie on a plane in $\mathbb{R}^3$, by B0, the sensors also lie on the same plane, and the localization problem can be thought of as localization in $\mathbb{R}^2$.

[4] Recall that the set $\mathcal{K}(l, r_l)$ groups the neighboring nodes of l within a radius r_l. By (6), $\Theta_l(r_l)$ is a subset of $m+1$ nodes such that sensor l lies in its convex hull but is not one of its elements.

To state the next assumption, define a communication link between nodes l and j, if l and j can exchange information. If l and j have a communication link, l and j can both estimate the internode distance, d_{lj}, between them. This distance can be found by received signal strength (RSS), time of arrival (ToA), or angle of arrival (AoA); see [27] for details. The estimate of this distance may be noisy. The discussion on the effects of noise and modifying DILOC such that it is robust to noise is deferred until Section V. We state the third assumption underlying DILOC.

B3) Internode Communication: There is a communication link between all of the nodes in the set $\{l\} \cup \mathcal{K}(l, r_l), \quad \forall l \in \Omega$.

With the above assumptions and notations, we present barycentric coordinates that serve as the updating coefficients in DILOC.

Barycentric Coordinates: DILOC is expressed in terms of the barycentric coordinates, a_{lk}, of the node, $l \in \Omega$, with respect to the nodes, $k \in \Theta_l$. The barycentric coordinates, a_{lk}, are unique and are given by (see [2] and [3])

$$a_{lk} = \frac{A_{\{l\} \cup \Theta_l \setminus \{k\}}}{A_{\Theta_l}}, \tag{7}$$

with $A_{\Theta_l} \neq 0$, where "\" denotes the set difference, and $A_{\{l\} \cup \Theta_l \setminus \{k\}}$ is the generalized volume of the set $\{l\} \cup \Theta_l \setminus \{k\}$, i.e., the set Θ_l with node l added and node k removed. The barycentric coordinates can be computed from the internode distances d_{lk} using the Cayley–Menger determinants as shown in Appendix II. From (7), and the facts that the generalized volumes are nonnegative and

$$\sum_{k \in \Theta_l} A_{\Theta_l \cup \{l\} \setminus \{k\}} = A_{\Theta_l}, \quad l \in \mathcal{C}(\Theta_l) \tag{8}$$

it follows that, for each $l \in \Omega, k \in \Theta_l$,

$$a_{lk} \in [0, 1], \quad \sum_{k \in \Theta_l} a_{lk} = 1. \tag{9}$$

We now present DILOC in two steps: setup and DILOC proper. We then provide its matrix form useful for analysis purposes.

DILOC Setup: Triangulation: In the setup step, each sensor l triangulates itself, so that by the end of this step we have paired every $l \in \Omega$ with its corresponding $m + 1$ neighbors in Θ_l. Since triangulation should be with a small r_l, the following is a practical protocol for the setup step.

Sensor l starts with a communication radius r_l that guarantees triangulation with high probability with the given density of deployment γ_0. This choice is explained in detail in Section II-D. Sensor l then chooses arbitrarily $m+1$ nodes within r_l and tests if it lies in the convex hull of these nodes using the procedure described in Appendix I. Sensor l attempts this with all collections of $m + 1$ nodes within r_l. If all attempts fail, the sensor adaptively increases, in small increments, its communication radius r_l and repeats the process.[5] By B0 and (4), success is eventually achieved, and each sensor is triangulated by finding Θ_l with properties (6) and B3.

If a sensor has directionality, a much simpler algorithm, based on Lemma 2 below (see also the discussion following the Lemma), triangulates the sensor with high probability of success in one shot. To assess the practical implications required by DILOC's setup phase, Section II-D considers the realistic scenario where the sensors are deployed using a random Poisson distribution and computes in terms of deployment parameters the probability of finding at least one such Θ_l in a given radius r_l.

DILOC Iterations: State Updating: Once the setup phase is complete, at time $t+1$, each sensor $l \in \Omega$ iteratively updates its state, i.e., its current location estimate, by a convex combination of the states at time t of the nodes in Θ_l. The anchors do not update their state, since they know their locations. The updating is explicitly given by

$$\mathbf{c}_l(t+1) = \begin{cases} \mathbf{c}_l(t), & l \in \kappa, \\ \sum_{k \in \Theta_l} a_{lk} \mathbf{c}_k(t), & l \in \Omega \end{cases} \tag{10}$$

where a_{lk} are the barycentric coordinates of l with respect to $k \in \Theta_l$. DILOC in (10) is distributed since i) the update is implemented at each sensor independently; ii) at sensor $l \in \Omega$, the update of the state, $\mathbf{c}_l(t+1)$, is obtained from the states of its $m+1$ neighboring nodes in Θ_l; and iii) there is no central location and only local information is available.

DILOC: Matrix Format: For compactness of notation and analysis purposes, we write DILOC (10) in matrix form. Without loss of generality, we index the anchors in κ as $1, 2, \ldots, m+1$ and the sensors in Ω as $m+2, m+3, \ldots, m+1+M = N$. We now stack the (row vectors) states $\mathbf{c}_l$, given in (3) for all the N nodes in the network in the $N \times m$-dimensional coordinate matrix

$$\mathbf{C} = \left[\mathbf{c}_1^T, \ldots, \mathbf{c}_N^T\right]^T. \tag{11}$$

DILOC equations in (10) now become in compact matrix form

$$\mathbf{C}(t+1) = \boldsymbol{\Upsilon} \mathbf{C}(t). \tag{12}$$

The structure of the $N \times N$ iteration matrix $\boldsymbol{\Upsilon}$ is more apparent if we partition it as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix}. \tag{13}$$

The first $m+1$ rows correspond to the update equations for the anchors in κ. Since the states of the anchors are constant, see B1 and (5), the first $m+1$ rows of $\boldsymbol{\Upsilon}$ are zero except for a 1 at their diagonal entry $(q, q), q \in \kappa = \{1, \ldots, m+1\}$. In other words, these first $m+1$ rows are the $(m+1) \times N$ block matrix $[\mathbf{I}_{m+1} \,|\, \mathbf{0}]$, i.e., the $(m+1) \times (m+1)$ identity matrix $\mathbf{I}_{m+1}$ concatenated with the $(m+1) \times M$ zero matrix $\mathbf{0}$.

Each of the M remaining rows in $\boldsymbol{\Upsilon}$, indexed by $l \in \Omega = \{m+2, m+3, \ldots, N\}$, have only $m+1$ nonzero elements corresponding to the nodes in the triangulation set Θ_l of l, and these nonzero elements are the barycentric coordinates a_{lk} of sensor l with respect to the nodes in Θ_l. The $M \times (m+1)$ block $\mathbf{B} = \{b_{lj}\}$ is a zero matrix, except in those entries b_{lj} corresponding to sensors $l \in \Omega$ that have a direct link to anchors $j \in \kappa$. The $M \times M$ block $\mathbf{P} = \{p_{lj}\}$ is also a sparse matrix where the nonzero entries in row l correspond to the sensors in

[5] The step size of this increment is also dependent on the density of the deployment, γ_0, such that a reasonable number of sensors are added in the larger neighborhood obtained by increasing r_l. This will be clear from the discussion in Section II-D.

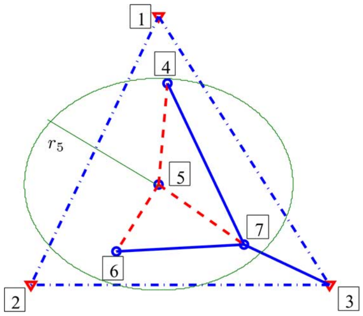

Fig. 1. Deployment corresponding to the example in Section II-C.

Θ_l. The matrices $\boldsymbol{\Upsilon}, \mathbf{P}$, and $\mathbf{B}$ have important properties that will be used to prove the convergence of the distributed iterative algorithm DILOC in Sections III and IV.

Remark: Equation (12) writes DILOC in matrix format for compactness; it should not be confused with a centralized algorithm—it still is a *distributed* iterative algorithm. It is iterative, because each iteration through (12) simply updates the (matrix of) state(s) from $\mathbf{C}(t)$ to $\mathbf{C}(t+1)$. It is distributed because each row equation updates the state of sensor l from a linear combination of the states of the $m+1$ nodes in Θ_l. In all, the iteration matrix, $\boldsymbol{\Upsilon}$, is highly sparse having exactly $(m+1)+M(m+1)$ nonzeros out of possible $(m+1+M)^2$ elements. Also, note that (12) is reminiscent of the consensus algorithm but the structure of the DILOC matrix in (13) and the goal of DILOC are very different from the corresponding objects in consensus. For a detailed study on the similarities among consensus, DILOC, and similar algorithms, see our recent work in [28] and [29] and the references therein.

C. Example

To illustrate DILOC, we consider an $m = 2$-dimensional Euclidean plane with $m+1 = 3$ anchors and $M = 4$ sensors; see Fig. 1. The nodes are indexed such that the anchor set is $\kappa = \{1,2,3\}, |\kappa| = m+1 = 3$, and the sensor set is $\Omega = \{4,5,6,7\}, |\Omega| = M = 4$. The set of all the nodes in the network is, thus, $\Theta = \kappa \cup \Omega = \{1,\ldots,7\}, |\Theta| = N = 7$. The triangulation sets $\Theta_l, l \in \Omega$ identified by using the convex hull inclusion test are $\Theta_4 = \{1,5,7\}, \Theta_5 = \{4,6,7\}, \Theta_6 = \{2,5,7\}, \Theta_7 = \{3,5,6\}$. These triangulation sets satisfy the properties in (6). Sensor 5 does not have any anchor node in its triangulation set Θ_5, while the other sensors have only one anchor in their respective triangulation sets. Since no sensor communicates with the $m+1 = 3$ anchors directly, no sensor can localize itself in a single step.

At each sensor, $l \in \Omega$, the barycentric coordinates $a_{lk}, k \in \Theta_l$ are computed using the internode distances (among the nodes in the set $\{l\} \cup \Theta_l$) in the Cayley–Menger determinant. It is noteworthy that the internode distances that need to be known at each sensor l to compute a_{lk} are only the internode distances among the $m+2$ nodes in the set $\{l\} \cup \Theta_l$. For instance, the distances in the Cayley–Menger determinant needed by sensor 5 to compute a_{54}, a_{56}, a_{57} are the internode distances among the nodes in the set $\{5\} \cup \Theta_5$, i.e., $d_{54}, d_{56}, d_{57}, d_{46}, d_{47}, d_{67}$. These internode distances are known at sensor 5 due to B3.

Once the barycentric coordinates a_{lk} are computed, DILOC for the sensors in Ω is

$$\mathbf{c}_l(t+1) = \sum_{k \in \Theta_l} a_{lk}\mathbf{c}_k(t), \quad l \in \Omega = \{4,5,6,7\}. \tag{14}$$

We write DILOC for this example in the matrix format (12):

$$\begin{bmatrix} \mathbf{c}_1(t+1) \\ \mathbf{c}_2(t+1) \\ \mathbf{c}_3(t+1) \\ \mathbf{c}_4(t+1) \\ \mathbf{c}_5(t+1) \\ \mathbf{c}_6(t+1) \\ \mathbf{c}_7(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ a_{41} & 0 & 0 & 0 & a_{45} & 0 & a_{47} \\ 0 & 0 & 0 & a_{54} & 0 & a_{56} & a_{57} \\ 0 & a_{62} & 0 & 0 & a_{65} & 0 & a_{67} \\ 0 & 0 & a_{73} & a_{74} & 0 & a_{76} & 0 \end{bmatrix} \times \begin{bmatrix} \mathbf{c}_1(t) \\ \mathbf{c}_2(t) \\ \mathbf{c}_3(t) \\ \mathbf{c}_4(t) \\ \mathbf{c}_5(t) \\ \mathbf{c}_6(t) \\ \mathbf{c}_7(t) \end{bmatrix} \tag{15}$$

where the initial conditions are $\mathbf{C}(0) = [\mathbf{c}_1^{*T}, \mathbf{c}_2^{*T}, \mathbf{c}_3^{*T}, \mathbf{c}_4{}^T(0), \mathbf{c}_5^T(0), \mathbf{c}_6^T(0), \mathbf{c}_7^T(0)]^T$, with $\mathbf{c}_l(0), l \in \Omega$, being randomly chosen row vectors of appropriate dimensions.

Note here again that (15) is just a matrix representation of (14). DILOC is implemented in a distributed fashion as in (14). The matrix representation in (15) is included for compaction of notation and for the convergence analysis of the algorithm. The sparseness in the matrix in (15) illustrates the locality of the communication among the nodes.

D. Random Poisson Deployment

For sensors to determine their locations, they need to triangulate. We first consider a sufficient condition for a sensor to triangulate. We illustrate it on the plane $m = 2$; the discussion can be extended to arbitrary dimensions. Consider Fig. 2(a), which shows the triangulation region, a circle of radius r_l centered at sensor l. Let Q_1, Q_2, Q_3, Q_4 be four disjoint sectors partitioning this circle with equal areas, i.e., $A_{Q_i} = (\pi r_l^2)/(4), i = 1,\ldots,4$.

Lemma 2: A sufficient condition to triangulate sensor $l \in \mathbb{R}^2$ is to have at least one node in each of the four disjoint equal area sectors $Q_i, i = 1,\ldots,4$, which partition the circle of radius r_l centered at l.

Proof: In Fig. 2(b), consider the triangulation of sensor l located at the center of the circle; we choose arbitrarily four nodes p_1, p_2, p_3, p_4 in each of the four sectors Q_1, Q_2, Q_3, Q_4. Denote the polygon with vertices p_1, p_2, p_3, p_4 by $\mathrm{Pol}(p_1p_2p_3p_4)$. Consider the diagonal[6] $p_1 - p_3$ that partitions

[6] If $\mathrm{Pol}(p_1p_2p_3p_4)$ is concave, we choose the diagonal that lies inside $\mathrm{Pol}(p_1p_2p_3p_4)$, i.e., $p_1 - p_3$ in Fig. 2(b). If $\mathrm{Pol}(p_1p_2p_3p_4)$ is convex, we can choose any of the two diagonals and the proof follows.

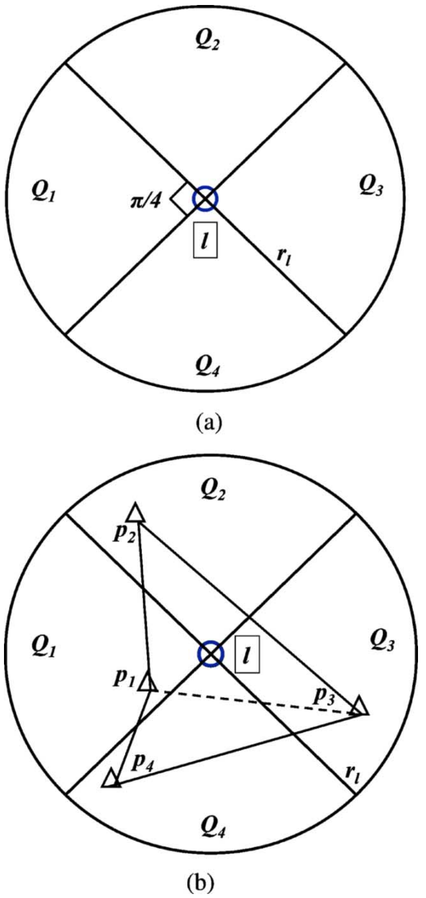

Fig. 2. (a) Sensor l identifies its triangulation set Θ_l in the circle of radius r_l centered at sensor l. The circle is divided into four disjoint sectors with equal areas $Q_1, \ldots, Q_4$. A sufficient condition for triangulation is that there exists at least one node in each of these four sectors. (b) Illustration of Lemma 2.

this polygon into two triangles $\triangle p_1p_2p_3$ and $\triangle p_1p_3p_4$. Since $l \in \text{Pol}(p_1p_2p_3p_4)$ and $\triangle p_1p_2p_3 \cup \triangle p_1p_3p_4 = \text{Pol}(p_1p_2p_3p_4)$ with $\triangle p_1p_2p_3 \cap \triangle p_1p_3p_4 = \text{LineSegment}(p_1, p_3)$, then either $l \in \triangle p_1p_2p_3$ or $l \in \triangle p_1p_3p_4$ or l belongs to both (when it falls on the $\text{LineSegment}(p_1, p_3)$). The triangle in which l lies becomes the triangulating set Θ_l of l.

This completes the proof. The generalization to higher dimensions is straightforward; for instance, in $\mathbb{R}^3$, we have eight sectors, and an arbitrary sensor l is triangulated with at least one node in each of these eight sectors (with equal volume) of a sphere of radius r_l centered around l. ■

In the following subsections, we study various probabilities associated with the triangulation, assuming a Poisson sensor deployment. For simplicity, we restrict the discussion to $m = 2$; it can be extended to arbitrary dimensions.

1) Probability That a Sensor Can Triangulate: Here, we provide a local result concerned with the triangulation of an arbitrary sensor. We characterize the probability that a sensor l can triangulate successfully in a region of radius r_l centered at l. A common deployment model in wireless sensor networks is the Poisson deployment [30], [31]. For a Poisson distribution with rate parameter or deployment density $\gamma_0 > 0$, the mean number of nodes in a sector Q with area A_Q is $\gamma_0 A_Q$. The numbers of nodes in any two disjoint sectors Q_1 and Q_2 are independent random variables, and the locations of the nodes in a sector Q are uniformly distributed. Let $\overline{Q}_i$ be the set of nodes in the sector Q_i. It follows from the Poisson deployment that the probability of finding at least one node in each of the four sets $\overline{Q}_1, \ldots, \overline{Q}_4$ is the product

$$\mathbb{P}(|\overline{Q}_i| > 0,\, 1 \le i \le 4) = (1 - e^{-\gamma_0 \pi r_l^2/4})^4 \tag{16}$$

since the distribution of the number of nodes in disjoint sectors is independent. Thus, we have

$$\begin{aligned}\mathbb{P}_{\Theta_l}(\gamma_0) &\triangleq \mathbb{P}(\{\text{sensor } l \text{ can triangulate}\})\\ &= \mathbb{P}(\Theta_l \text{ exists satisfying (6) given } \gamma_0)\\ &\ge \mathbb{P}(|\overline{Q}_i| > 0, 1 \le i \le 4).\end{aligned} \tag{17}$$

The probability that a sensor fails to triangulate is

$$\begin{aligned}\mathbb{P}_{F,l}(\gamma_0) &\triangleq \mathbb{P}(\{\text{sensor } l \text{ cannot triangulate}\})\\ &= 1 - \mathbb{P}_{\Theta_l}(\gamma_0).\end{aligned} \tag{18}$$

Equations (17) and (18) provide a tradeoff between r_l and γ_0, Indeed, to guarantee triangulation of sensor l with probability ϵ, arbitrarily close to 1, either

$$\begin{aligned} r_l &\ge \left(\frac{-4ln\left(1-\epsilon^{\frac{1}{4}}\right)}{\gamma_0\pi}\right)^{\frac{1}{2}}, \quad \text{or}\\ \gamma_0 &\ge \frac{-4}{\pi r_l^2} ln\left(1-\epsilon^{\frac{1}{4}}\right).\end{aligned} \tag{19}$$

2) Probability That All M Sensors Triangulate: Here, we provide a global result, i.e., we (lower) bound the probability that all sensors in the network triangulate. We have

$$\begin{aligned}&\mathbb{P}(\{\text{sensor } l \text{ triangulates}\}, 1 \le l \le M)\\ &\quad = 1 - \mathbb{P}\left(\bigcup_l \{\text{sensor } l \text{ cannot triangulate}\}\right)\\ &\quad \ge 1 - \sum_l \mathbb{P}(\{\text{sensor } l \text{ cannot triangulate}\})\\ &\quad = 1 - M\mathbb{P}_{F,l}(\gamma_0)\end{aligned} \tag{20}$$

where we use the union bound to go from the first equation to the second. To get the third equation, we use (18) and assume a flat network, i.e., a network where all the sensors have the same characteristics (in particular, r_l is the same for all sensors). Clearly, the bound in (20) is only meaningful if $\mathbb{P}_{F,l}(\gamma_0)$ is very small.

3) Probability That the Resulting Sensor Network Triangulates Given Triangulation Failures: Given that some sensors may fail to triangulate, we ask the question of what is the probability that the remaining sensors can all triangulate. An exact expression is beyond the scope of this paper. Here, we give a plausible argument when the number of sensors is large so that the law of large numbers is valid. Since the probability of failure of each sensor to triangulate is $\mathbb{P}_{F,l}$ in (18), $M\mathbb{P}_{F,l}$ sensors fail to triangulate. Hence, to compute the probability that the reduced network (the network of sensors that can triangulate once we exclude the sensors that could not) triangulates, we simply repeat the steps in Sections II-D-1 and II-D-2, but, now with M_1 sensors and deployment density γ_1, given by

$$M_1 = M(1 - \mathbb{P}_{F,l}(\gamma_0)) \tag{21}$$

$$\gamma_1 = \gamma_0(1 - \mathbb{P}_{F,l}(\gamma_0)). \tag{22}$$

From (20), the probability that the reduced network triangulates is

$$\begin{aligned}\mathbb{P}(&\text{sensor } l \text{ triangulates}, \forall l)\\ &\geq 1 - M_1 \mathbb{P}_{F,l}(\gamma_1), && (23)\\ &= 1 - M(1 - \mathbb{P}_{F,l}(\gamma_0))\mathbb{P}_{F,l}(\gamma_1). && (24)\end{aligned}$$

Equation (24) shows that, although not all sensors can triangulate, the probability of triangulating the reduced network can be made arbitrarily high by choosing either γ_0, or r_l, or both appropriately, such that $\mathbb{P}_{F,l}(\gamma_1) \rightarrow 0$ (or alternatively $\mathbb{P}_{\Theta_l}(\gamma_1) \rightarrow 1$).

E. Complexity of DILOC

Once the barycentric coordinates are computed, each sensor performs the update in (10) that requires $m+1$ multiplications and m additions. Assuming the computation complexity of the multiplication and the addition operations to be the same, the computation complexity of DILOC is $2m+1$ operations, i.e., $O(1)$ per sensor, per iteration. Since each sensor exchanges information with $m+1$ nodes in its triangulation set, the communication complexity of DILOC is $m+1$ communications, i.e., $O(1)$ per sensor, per iteration. Hence, both the computation and communication complexity are $O(M)$ for a network of M sensors. Note that, since the triangulation setup phase,[7] which identifies $\Theta_l(r_l)$ and computes the barycentric coordinates, as explained in Section II-B, are to be carried out only once at the start of DILOC, they require a constant computation/communication complexity, so we do not account for it explicitly.

III. Convergence of DILOC: Deterministic Scenario

In this section, we prove the convergence of DILOC to the exact locations of the sensors, $\mathbf{c}_l^*, l \in \Omega$, when there is no randomness, i.e., the communication is noiseless, required distances are known precisely, and the communication links are active all the time (no packet losses). Distributed localization in random environments is the subject of Sections V and VI. To formally state the convergence result, we provide, briefly, background and additional results, based on assumptions B0–B3.

The entries of the rows of the iteration matrix $\boldsymbol{\Upsilon}$, in (12), are either zero or the barycentric coordinates a_{lk}, which are nonnegative, and, by (9), add to 1. This matrix can then be interpreted as the transition matrix of a Markov chain. We then describe the localization problem and DILOC in terms of a Markov chain. Let the assumptions B0–B3 in Section II-B hold, and the N nodes in the sensor network correspond to the states of a Markov chain where the (ij)th element of the iteration matrix $\boldsymbol{\Upsilon} = \{\upsilon_{ij}\}$ defines the probability that the ith state goes to the jth state. Because of the structure of $\boldsymbol{\Upsilon}$, this chain is a very special Markov chain.

Absorbing Markov Chain: Let an $N \times N$ matrix, $\boldsymbol{\Upsilon} = \{\upsilon_{ij}\}$, denote the transition probability matrix of a Markov chain with N states $s_{i,i=1,\ldots,N}$. A state s_i is called absorbing if the probability of leaving that state is 0 (i.e., $\upsilon_{ij} = 0, i \neq j$, in other words $\upsilon_{ii} = 1$). A Markov chain is said to be absorbing if it has at least one absorbing state, and if from every state it is possible to go with nonzero probability to an absorbing state (not necessarily in one step). In an absorbing Markov chain, a state that is not absorbing is called transient. For additional background, see, for example, [32].

Lemma 3: The underlying Markov chain with the transition probability matrix given by the iteration matrix $\boldsymbol{\Upsilon}$ is absorbing.

Proof: We prove by contradiction. Since $\upsilon_{ii} = 1, i \in \kappa$, the anchors are the absorbing states of the Markov chain. We now show that the Markov chain is absorbing with the sensors as the transient states.

Assume that the underlying Markov chain is not absorbing. This can happen only if the transient states can be partitioned into two disjoint clusters C1 and C2 (with C2 nonempty), such that each nonabsorbing state (sensor) in C1 can reach, with nonzero probability, at least one of the absorbing states (i.e., there is a directed path from each nonabsorbing state to at least one of the anchors) and, with probability 1, the states in C2 cannot reach an absorbing state (i.e., there is no directed path from the transient state to any anchor). It follows with probability 1 that the states in C2 cannot reach the states in C1 (in one or multiple steps); otherwise, they reach an absorbing state with a nonzero probability.

Now consider the nonabsorbing states (or sensors) that lie on the boundary of the convex hull $\mathcal{C}$(C2), i.e., the vertices of $\mathcal{C}$(C2). Because they are on the boundary, they cannot lie in the interior of the convex hull of any subset of sensors in $\mathcal{C}$(C2), and, thus, cannot triangulate themselves, which contradicts Lemma 1. In order to triangulate the boundary sensors in $\mathcal{C}$(C2), the boundary sensors in C2 must be able to reach the nonabsorbing states (sensors) of C1 and/or the absorbing states (anchors); that is to say that the boundary sensors in $\mathcal{C}$(C2) have directed paths to the absorbing states (anchors). This clearly contradicts the assumption that the set C2 is nonempty and, hence, every nonabsorbing state has a directed path to the absorbing states. Hence, the Markov chain is absorbing and the sensors correspond to the transient states of the absorbing Markov chain. ■

Consider the partitioning of the iteration matrix, $\boldsymbol{\Upsilon}$, in (13). With the Markov chain interpretation, the $M \times (m+1)$ block $\mathbf{B} = \{b_{lj}\}$ is a transition probability matrix for the transient states to reach the absorbing states in one step, and the $M \times M$ block $\mathbf{P} = \{p_{lj}\}$ is a transition probability matrix for the transient states. With (13), $\boldsymbol{\Upsilon}^{t+1}$ can be written as

$$\boldsymbol{\Upsilon}^{t+1} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \sum_{k=0}^{t} \mathbf{P}^k \mathbf{B} & \mathbf{P}^{t+1} \end{bmatrix} \tag{25}$$

[7] It follows from Lemma 2 that if the sensors have directional capability then each sensor has to find one neighbor in each of its four sectors $Q_{l,1}, Q_{l,2}, Q_{l,3}, Q_{l,4}$ (in $m = 2$-dimensional space). Once a neighbor is found, triangulation requires choosing three out of these four, in order to identify Θ_l. The computational complexity in $m = 2$-dimensional Euclidean space is $4 \text{ choose } 3 = 4$. Without directionality, the process of finding Θ_l has the (expected) computation complexity of $\gamma_0 \pi r_l^2 \text{ choose } 3$.

and, as t goes to infinity, we have

$$\lim_{t\to\infty} \boldsymbol{\Upsilon}^{t+1} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B} & \mathbf{0} \end{bmatrix} \tag{26}$$

by Lemmas 5 and 6, in Appendix III. Lemmas 5 and 6 use the fact that if $\mathbf{P}$ is the matrix associated to the transient states of an absorbing Markov chain, then $\rho(\mathbf{P}) < 1$, where $\rho(\,\cdot\,)$ is the spectral radius of a matrix. With (26), DILOC (10) converges to

$$\lim_{t\to\infty} \mathbf{C}(t+1) = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B} & \mathbf{0} \end{bmatrix} \mathbf{C}(0). \tag{27}$$

From (27), we note that the coordinates of the M sensors in Ω (last M rows of $\mathbf{C}(t+1)$) converge as $t \to \infty$ to functions of the $m+1$ anchors in κ (whose coordinates are exactly known). The limiting values of the states of the M sensors in Ω are written in terms of the coordinates of the $m+1$ anchors in κ weighted by $(\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B}$. To show that the limiting values are indeed the exact solution, we give the following lemma.

Lemma 4: Let $\mathbf{c}_l^*$ be the exact coordinates of a node $l \in \Theta$. Let the $M \times (m+1)$ matrix $\mathbf{D} = \{d_{lj}\}, l \in \Omega, j \in \kappa$ be the matrix of the barycentric coordinates of the M sensors (in Ω) in terms of the $m+1$ anchors in κ, relating the coordinates of the sensors to the coordinates of the anchors by

$$\mathbf{c}_l^* = \sum_{j\in\kappa} d_{lj}\mathbf{c}_j^*, \quad l \in \Omega. \tag{28}$$

Then, we have

$$\mathbf{D} = (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B}. \tag{29}$$

Proof: Clearly $(\mathbf{I}_M - \mathbf{P})$ is invertible, since, by (100) in Appendix III, $\rho(\mathbf{P}) < 1$; this follows from the fact that the eigenvalues of the matrix $\mathbf{I}_M - \mathbf{P}$ are $1 - \lambda_j$, where λ_j is the jth eigenvalue of the matrix $\mathbf{P}$ and $|\lambda_j| < 1, \quad \forall j = 1, \ldots, M$. It suffices to show that

$$\mathbf{D} = \mathbf{B} + \mathbf{P}\mathbf{D} \tag{30}$$

since (29) follows from (30). In (30), $\mathbf{D}$ and $\mathbf{B}$ are both $M \times (m+1)$ matrices, whereas $\mathbf{P}$ is an $M \times M$ matrix whose nonzero elements are the barycentric coordinates for the sensors in Ω. Hence, for the ljth element in (30), we need to show that

$$d_{lj} = b_{lj} + \sum_{k\in\Omega} p_{lk}d_{kj}. \tag{31}$$

For an arbitrary sensor, $l \in \Omega$, its triangulation set, Θ_l, may contain nodes from both κ and Ω. We denote κ_{Θ_l} as the elements of Θ_l that are anchors, and Ω_{Θ_l} as the elements of Θ_l that are sensors, i.e., nonanchors. The exact coordinates $\mathbf{c}_l^*$ of the sensor l can be expressed as a convex combination of the coordinates of its neighbors in its triangulation set $k \in \Theta_l$, using the barycentric coordinates a_{lk}, i.e.,

$$\begin{aligned} \mathbf{c}_l^* &= \sum_{k\in\Theta_l} a_{lk}\mathbf{c}_k^* \\ &= \sum_{j\in\kappa_{\Theta_l}} a_{lj}\mathbf{c}_j^* + \sum_{k\in\Omega_{\Theta_l}} a_{lk}\mathbf{c}_k^* \\ &= \sum_{j\in\kappa} b_{lj}\mathbf{c}_j^* + \sum_{k\in\Omega} p_{lk}\mathbf{c}_k^* \end{aligned} \tag{32}$$

since the scalars a_{lj} are given by

$$a_{lj} = \begin{cases} b_{lj}, & \text{if } j \in \kappa_{\Theta_l} \\ p_{lj}, & \text{if } j \in \Omega_{\Theta_l} \\ 0, & \text{if } j \notin \Theta_l. \end{cases} \tag{33}$$

Equation (32) becomes, after writing each $k \in \Omega$ in terms of the $m+1$ anchors in κ,

$$\begin{aligned} \mathbf{c}_l^* &= \sum_{j\in\kappa} b_{lj}\mathbf{c}_j^* + \sum_{k\in\Omega} p_{lk} \sum_{j\in\kappa} d_{kj}\mathbf{c}_j^*, \\ &= \sum_{j\in\kappa} b_{lj}\mathbf{c}_j^* + \sum_{j\in\kappa}\sum_{k\in\Omega} p_{lk}d_{kj}\mathbf{c}_j^*, \\ &= \sum_{j\in\kappa}\left(b_{lj} + \sum_{k\in\Omega} p_{lk}d_{kj}\right)\mathbf{c}_j^*. \end{aligned} \tag{34}$$

This is a representation of the coordinates of sensor, l, in terms of the coordinates of the anchors, $j \in \kappa$. Since for each $j \in \kappa$, the value inside the parentheses is nonnegative with their sum over $j \in \kappa$ being 1, and the fact that the barycentric representation is unique, we must have

$$d_{lj} = b_{lj} + \sum_{k\in\Omega} p_{lk}d_{kj} \tag{35}$$

which, comparing to (28), completes the proof. ■

We now recapitulate these results in the following theorem.

Theorem 1 (DILOC Convergence): DILOC (10) converges to the *exact* sensor coordinates, $\mathbf{c}_l^*, l \in \Omega$, i.e.,

$$\lim_{t\to\infty} \mathbf{c}_l(t+1) = \mathbf{c}_l^* \quad \forall l \in \Omega. \tag{36}$$

Proof: The proof is a consequence of Lemmas 3 and 4.

Convergence Rate: The convergence rate of the localization algorithm depends on the spectral radius of the matrix $\mathbf{P}$, which by (100) in Appendix III is strictly less than one. In fact, using standard matrix arguments, one can show that DILOC is characterized by geometric convergence rate with exponent $\rho(\mathbf{P})$. This is a consequence of the fact that $\mathbf{P}$ is a uniformly substochastic matrix. The convergence is slow if the spectral radius $\rho(\mathbf{P})$ is close to 1. This can happen if the matrix $\mathbf{B}$ is close to a zero matrix $\mathbf{0}$. This is the case if the sensors cluster in a region of very small area inside the convex hull of the anchors, and the anchors themselves are very far apart. In fact, it can be seen that in this case the barycentric coordinates for the sensors with $\kappa_{\Theta_l} \neq \emptyset$ (see Lemma 4 for this notation) corresponding to the elements in κ_{Θ_l} are close to zero. When, as in practical wireless sensor applications, the nodes are assumed to be deployed in a geometric or a Poisson fashion (see details in Section II-D), the above event is highly improbable.

IV. DILOC With Relaxation

In this Section, we modify DILOC to obtain a form that is more suitable to study distributed localization in random environments. We observe that in DILOC (10), at time $t+1$, the expression for $\mathbf{c}_l(t+1), l \in \Omega$ does not involve its own coordinates $\mathbf{c}_l(t)$ at time t. We introduce a relaxation parameter $\alpha \in (0,1]$ in the iterations, such that the expression of $\mathbf{c}_l(t+1)$ is a convex combination of $\mathbf{c}_l(t)$ and (10). We refer to this version as DILOC *with relaxation* (DILOC-REL). It is given by

$$\mathbf{c}_l(t+1) = \begin{cases} (1-\alpha)\mathbf{c}_l(t) + \alpha\mathbf{c}_l(t) = \mathbf{c}_l(t), & l \in \kappa \\ (1-\alpha)\mathbf{c}_l(t) + \alpha\sum_{k\in\Theta_l} a_{lk}\mathbf{c}_k(t), & l \in \Omega. \end{cases} \tag{37}$$

DILOC is the special case of DILOC-REL with $\alpha = 1$. Clearly, DILOC-REL is also distributed as the sensor updates now have additional terms corresponding to their own states. The matrix representation of DILOC-REL is

$$\mathbf{C}(t+1) = \mathbf{H}\mathbf{C}(t) \tag{38}$$

where $\mathbf{H} = (1-\alpha)\mathbf{I}_N + \alpha\mathbf{\Upsilon}$ and $\mathbf{I}_N$ is the $N \times N$ identity matrix. It is straightforward to show that the iteration matrix $\mathbf{H}$ corresponds to a transition probability matrix of an absorbing Markov chain, where the anchors are the absorbing states and the sensors are the transient states. Let $\mathbf{J} = (1-\alpha)\mathbf{I}_M + \alpha\mathbf{P}$, partitioning $\mathbf{H}$ as

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \alpha\mathbf{B} & \mathbf{J} \end{bmatrix}. \tag{39}$$

We note the following:

$$\mathbf{H}^{t+1} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \sum_{k=0}^{t}\mathbf{J}^k\alpha\mathbf{B} & \mathbf{J}^{t+1} \end{bmatrix} \tag{40}$$

$$\lim_{t\to\infty}\mathbf{H}^{t+1} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ (\mathbf{I}_M - \mathbf{J})^{-1}\alpha\mathbf{B} & \mathbf{0} \end{bmatrix} \tag{41}$$

from Lemmas 5 and 6. Lemmas 5 and 6 apply to $\mathbf{H}$, since $\mathbf{H}$ is nonnegative and $\rho(\mathbf{J}) < 1$. To show $\rho(\mathbf{J}) < 1$, we recall that $\rho(\mathbf{P}) < 1$ and the eigenvalues of $\mathbf{J}$ are $(1-\alpha) + \alpha\lambda_j$, where λ_j are the eigenvalues of $\mathbf{P}$. Therefore, we have

$$\rho(\mathbf{J}) = \max_j |(1-\alpha) + \alpha\lambda_j| < 1. \tag{42}$$

The following theorem establishes convergence of DILOC-REL.

Theorem 2: DILOC-REL (37) converges to the *exact* sensor coordinates, $\mathbf{c}_l^*, l \in \Omega$, i.e.,

$$\lim_{t\to\infty}\mathbf{c}_l(t+1) = \mathbf{c}_l^* \quad \forall l \in \Omega. \tag{43}$$

Proof: It suffices to show that

$$(\mathbf{I}_M - \mathbf{J})^{-1}\alpha\mathbf{B} = (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B}. \tag{44}$$

To this end, we note that

$$(\mathbf{I}_M - \mathbf{J})^{-1}\alpha\mathbf{B} = (\mathbf{I}_M - ((1-\alpha)\mathbf{I}_M + \alpha\mathbf{P}))^{-1}\alpha\mathbf{B} \tag{45}$$

which reduces to (44) and the convergence of DILOC-REL follows from Lemma 4. ■

As mentioned, DILOC-REL is the basis for the distributed localization algorithm in random environments (DLRE) that we discuss in Sections V and VI.

V. Distributed Localization in Random Environments: Assumptions and Algorithm

This and the next section study distributed iterative localization in more realistic practical scenarios, when the internode distances are known up to errors, the communication links between nodes may fail, and, when alive, the communication among nodes is corrupted by noise. We write the update equations for DILOC-REL, (38), in terms of the columns $\mathbf{c}^j(t), 1 \le j \le m$ of the coordinate matrix $\mathbf{C}(t)$. Column j corresponds to the vector of the jth estimated coordinates of all the N nodes.[8] The updates are

$$\mathbf{c}^j(t+1) = [(1-\alpha)\mathbf{I} + \alpha\mathbf{\Upsilon}]\,\mathbf{c}^j(t), \qquad 1 \le j \le m. \tag{46}$$

We partition $\mathbf{c}^j(t)$ as

$$\mathbf{c}^j(t) = \begin{bmatrix} \mathbf{u}^j \\ \mathbf{x}^j(t) \end{bmatrix} \tag{47}$$

where, $\mathbf{u}^j \in \mathbb{R}^{(m+1)\times 1}$ corresponds to the jth coordinates of the anchors, which are known (hence, we omit the time index), and $\mathbf{x}^j(t) \in \mathbb{R}^{M\times 1}$ corresponds to the estimates of the jth coordinates of the sensors, hence not known. Since the update is performed only on the $\mathbf{x}^j(t)$, (46) is equivalent to the following recursion:

$$\mathbf{x}^j(t+1) = [(1-\alpha)\mathbf{I} + \alpha\mathbf{P}]\mathbf{x}^j(t) + \alpha\mathbf{B}\mathbf{u}^j. \tag{48}$$

Thus, to implement the sequence of iterations in (48) perfectly, the lth sensor at iteration t needs the corresponding rows of the matrices $\mathbf{P}$ and $\mathbf{B}$, and, in addition, the current estimates $c_n^j(t), n \in \Theta_l$ (the jth component of the nth sensor coordinates) of its neighbors' positions. In practice, there are several limitations: i) The computation of the matrices $\mathbf{P}$ and $\mathbf{B}$ requires internode distance computations, which are not perfect in a random environment; ii) the communication channels, or links, between neighboring channels may fail at random times; and iii) because of imperfect communication, each sensor receives only noisy versions of its neighbors' current state. Hence, in a random environment, we need to modify the iteration sequence in (48) to account for the partial imperfect information received by a sensor at each iteration. We start by stating formally our modeling assumptions.

C1) Randomness in System Matrices: At each iteration, each sensor needs the corresponding row of the system matrices $\mathbf{B}$

[8] In the sequel, we omit the subscripts from the identity matrix, $\mathbf{I}$, and its dimensions will be clear from the context.

and $\mathbf{P}$, which, in turn, depend on the internode distance measurements and can be, possibly, random. Since a single measurement of the internode distances may lead to a large random noise, we assume the sensors estimate the required distances at each iteration of the algorithm (note that this leads to an implicit averaging of the unbiased noisy effects, as will be demonstrated later). In other words, at each iteration, the lth sensor can only get estimates $\hat{\mathbf{B}}_l(t)$ and $\hat{\mathbf{P}}_l(t)$ of the corresponding rows of the $\mathbf{B}$ and $\mathbf{P}$ matrices, respectively. In the generic imperfect communication case, we have

$$\widehat{\mathbf{B}}(t) = \mathbf{B} + \mathbf{S}_{\mathbf{B}} + \widetilde{\mathbf{S}}_{\mathbf{B}}(t) \tag{49}$$

where $\{\widetilde{\mathbf{S}}_{\mathbf{B}}(t)\}_{t\geq 0}$ is an independent sequence of random matrices with

$$\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{B}}(t)] = 0, \forall t, \quad \sup_{t\geq 0} \mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{B}}(t)\|^2] = k_{\mathbf{B}} < \infty. \tag{50}$$

Here, $\mathbf{S}_{\mathbf{B}}$ is the mean measurement error. Similarly, for $\mathbf{P}$, we have

$$\widehat{\mathbf{P}}(t) = \mathbf{P} + \mathbf{S}_{\mathbf{P}} + \widetilde{\mathbf{S}}_{\mathbf{P}}(t) \tag{51}$$

where $\{\widetilde{\mathbf{S}}_{\mathbf{P}}(t)\}_{t\geq 0}$ is an independent sequence of random matrices with

$$\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{P}}(t)] = 0, \forall t, \quad \sup_{t\geq 0} \mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{P}}(t)\|^2] = k_{\mathbf{P}} < \infty. \tag{52}$$

Likewise, $\mathbf{S}_{\mathbf{P}}$ is the mean measurement error. Note that this way of writing $\widehat{\mathbf{B}}(t), \widehat{\mathbf{P}}(t)$ does not require the noise model to be additive. It only says that any random object may be written as the sum of a deterministic mean part and the corresponding zero mean random part. The moment assumptions in (50) and (52) are very weak and, in particular, are satisfied if the sequences $\{\widehat{\mathbf{B}}(t)\}_{t\geq 0}$ and $\{\widehat{\mathbf{P}}(t)\}_{t\geq 0}$ are i.i.d. with finite variance.

C2) Random Link Failure: We assume that the internode communication links fail randomly. This happens, for example, in wireless sensor network applications, where occasionally data packets are dropped. To this end, if the sensors l and n share a communication link (or, $n \in \Theta_l$), we assume that the link fails with some probability $1 - q_{ln}$ at each iteration, where $0 < q_{ln} \leq 1$. We associate with each such potential network link a binary random variable $e_{ln}(t)$, where $e_{ln}(t) = 1$ indicates that the corresponding network link is active at time t, whereas $e_{ln}(t) = 0$ indicates a link failure. Note that $\mathbb{E}[e_{ln}] = q_{ln}$.

C3) Additive Channel Noise: Let $\{v_{ln}^j(t)\}_{l,n,j,t}$ be a family of independent zero mean random variables such that

$$\sup_{l,n,j,t} \mathbb{E}\left[v_{ln}^j(t)\right]^2 = k_v < \infty. \tag{53}$$

We assume that, at the tth iteration, if the network link (l, n) is active, sensor l receives only a corrupt version $y_{ln}^j(t)$ of sensor n's state $c_n^j(t)$, given by

$$y_{ln}^j(t) = c_n^j(t) + v_{ln}^j(t). \tag{54}$$

This models the channel noise. The moment assumption in (53) is very weak and holds, in particular, if the channel noise is i.i.d. with finite variance.

C4) Independence: We assume that the sequences $\{\widetilde{\mathbf{S}}_{\mathbf{B}}(t), \widetilde{\mathbf{S}}_{\mathbf{P}}(t)\}_{t\geq 0}, \{e_{ln}(t)\}_{l,n,t}$, and $\{v_{ln}^j(t)\}_{l,n,j,t}$ are mutually independent. These assumptions do not put restrictions on the distributional form of the random errors, only that they obey some weak moment conditions.

Under the random environment model [Assumptions C1–C4], the algorithm in (48) is not appropriate to update the sensors states. We consider the following state update recursion for the random environment case.

Distributed Localization in Random Environment Algorithm (DLRE):

$$\begin{aligned} x_l^j&(t+1) \\ &= (1-\alpha(t))x_l^j(t) \\ &\quad + \alpha(t)\left[\sum_{n\in\kappa\cap\Theta_l} \frac{e_{ln}(t)\widehat{\mathbf{B}}_{ln}(t)}{q_{ln}}\left(u_n^j + v_{ln}^j(t)\right)\right] \\ &\quad + \alpha(t)\left[\sum_{n\in\Omega\cap\Theta_l} \frac{e_{ln}(t)\widehat{\mathbf{P}}_{ln}(t)}{q_{ln}}\left(x_n^j(t) + v_{ln}^j(t)\right)\right], \\ &\qquad\qquad l\in\Omega, 1\leq j\leq m. \end{aligned} \tag{55}$$

In contrast with DILOC-REL, in (55) the gain $\alpha(t)$ is now time varying. It will become clear why when we study the convergence of this algorithm. To write DLRE in a compact form, we introduce notation. Define the random matrices, $\widetilde{\mathbf{B}}(t) \in \mathbb{R}^{M\times(m+1)}$ and $\widetilde{\mathbf{P}}(t) \in \mathbb{R}^{M\times(m+1)}$, as the matrices with ln entries given by

$$\begin{aligned} \widetilde{\mathbf{B}}_{ln}(t) &= \widehat{\mathbf{B}}_{ln}(t)\left(\frac{e_{ln}(t)}{q_{ln}} - 1\right) \\ \widetilde{\mathbf{P}}_{ln}(t) &= \widehat{\mathbf{P}}_{ln}(t)\left(\frac{e_{ln}(t)}{q_{ln}} - 1\right). \end{aligned} \tag{56}$$

Clearly, by C2 and C4, the matrices $\widetilde{\mathbf{B}}(t) \in \mathbb{R}^{M\times(m+1)}$ and $\widetilde{\mathbf{P}}(t) \in \mathbb{R}^{M\times(m+1)}$ are zero mean. Note that $\mathbb{E}[e_{ln}] = q_{ln}$. Also, by the bounded moment assumptions in C1, we have

$$\begin{aligned} &\sup_{t\geq 0} \mathbb{E}[\|\widetilde{\mathbf{B}}(t)\|^2] = \tilde{k}_{\mathbf{B}} < \infty \\ &\sup_{t\geq 0} \mathbb{E}[\|\widetilde{\mathbf{P}}(t)\|^2] = \tilde{k}_{\mathbf{P}} < \infty. \end{aligned} \tag{57}$$

Hence, the iterations in (55) can be written in vector form as

$$\begin{aligned} \mathbf{x}^j(t+1) = (1-\alpha(t))\mathbf{x}^j(t) + \alpha(t)[(&\widehat{\mathbf{P}}(t) \\ +\widetilde{\mathbf{P}}(t))\mathbf{x}^j(t) + (&\widehat{\mathbf{B}}(t) + \widetilde{\mathbf{B}}(t))\mathbf{u}^j + \boldsymbol{\eta}^j(t)] \end{aligned} \tag{58}$$

where the lth element of the vector $\boldsymbol{\eta}^j(t)$ is given by

$$\begin{aligned} \boldsymbol{\eta}_l^j(t) = &\sum_{n\neq l}(\widehat{\mathbf{P}}_{ln}(t) + \widetilde{\mathbf{P}}_{ln}(t))v_{ln}^j(t) \\ &+ \sum_{n\neq l}(\widehat{\mathbf{B}}_{ln}(t) + \widetilde{\mathbf{B}}_{ln}(t))v_{ln}^j(t). \end{aligned} \tag{59}$$

By C1–C4, the sequence $\{\boldsymbol{\eta}^j(t)\}_{t\geq 0}$ is zero mean, independent, with

$$\sup_t \mathbb{E}[\|\boldsymbol{\eta}^j(t)\|^2] = k_\eta < \infty. \tag{60}$$

From C1, the iteration sequence in (58) can be written as

$$\begin{aligned}\mathbf{x}^j(t+1) = \mathbf{x}^j(t) - \alpha(t)[&(\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})\mathbf{x}^j(t)\\ &- (\mathbf{B}+\mathbf{S}_\mathbf{B})\mathbf{u}^j - (\widetilde{\mathbf{S}}_\mathbf{P}(t)+\widetilde{\mathbf{P}}(t))\mathbf{x}^j(t).\\ &- (\widetilde{\mathbf{S}}_\mathbf{B}(t)+\widetilde{\mathbf{B}}(t))\mathbf{u}^j - \boldsymbol{\eta}^j(t)].\end{aligned} \tag{61}$$

We now make two additional design assumptions.

D1) Persistence Condition: The weight sequence satisfies

$$\alpha(t) > 0, \sum_{t\geq 0}\alpha(t) = \infty, \sum_{t\geq 0}\alpha^2(t) < \infty. \tag{62}$$

By this condition, common in adaptive control and signal processing, the weights decay to zero, but not too fast.

D2) Low Error Bias: We assume that

$$\rho(\mathbf{P}+\mathbf{S}_\mathbf{P}) < 1. \tag{63}$$

Clearly, $\rho(\mathbf{P}) < 1$. Thus, if the nonzero bias $\mathbf{S}_\mathbf{P}$ in the system matrix (resulting from incorrect distant computation) is small, (63) is justified. This condition ensures that the matrix $(\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})$ is invertible. In the next sections, we prove that, under C1–C4, D1–D2, DLRE's state vector sequence $\{\mathbf{x}^j(t)\}_{t\geq 0}$, a.s. converges to a deterministic vector for each j, possibly different from the exact sensor locations, due to the random errors in the iterations. We characterize this error and show that it depends on the nonzero biases $\mathbf{S}_\mathbf{B}$ and $\mathbf{S}_\mathbf{P}$ in the system matrix computations and vanishes as $\|\mathbf{S}_\mathbf{B}\| \to 0$ and $\|\mathbf{S}_\mathbf{P}\| \to 0$.

VI. DLRE: A.S. Convergence

We show the almost sure convergence of DLRE under the random environment presented in Section V.

Theorem 3 (DLRE a.s. Convergence): Let $\{\mathbf{x}^j(t)\}_{t\geq 0}, 1 \leq j \leq m$, be the state sequence generated by the iterations, given by (61), under the assumptions C1–C4, D1–D2. Then

$$\mathbf{P}\left[\lim_{t\to\infty}\mathbf{x}^j(t) = (\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})^{-1}(\mathbf{B}+\mathbf{S}_\mathbf{B})\mathbf{u}^j, \forall j\right] = 1. \tag{64}$$

DLRE's convergence analysis is based on the sample path properties of controlled Markov processes, which has also been used recently to prove convergence properties of distributed iterative stochastic algorithms in sensor networks, e.g., [33] and [34]. The proof relies on the following result [35] stated as a theorem.

Theorem 4: Consider the following recursive procedure:

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \alpha(t)[\mathbf{R}(\mathbf{x}(t)) + \boldsymbol{\Gamma}(t+1, \mathbf{x}(t), \omega)] \tag{65}$$

where $\mathbf{x}, \mathbf{R}, \boldsymbol{\Gamma}$ are vectors in $\mathbb{R}^{M\times 1}$. There is an underlying common probability space $(\Xi, \mathcal{F}, \mathbf{P})$; let ω be the canonical element of the probability space Ξ. Assume that the following conditions are satisfied.[9]

1) The vector function $\mathbf{R}(\mathbf{x})$ is Borel measurable and $\boldsymbol{\Gamma}(t, \mathbf{x}, \omega)$ is $\mathcal{B}^M \otimes \mathcal{F}$ measurable for every t.

2) There exists a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ of $\mathcal{F}$, such that the family of random vectors $\boldsymbol{\Gamma}(t, \mathbf{x}, \omega)$ is $\mathcal{F}_t$ measurable, zero-mean and independent of $\mathcal{F}_{t-1}$.

3) There exists a function $V(\mathbf{x}) \in \mathbb{C}^2$ with bounded second order partial derivatives satisfying:

$$V(\mathbf{x}_0) = 0, V(\mathbf{x}) > 0, \mathbf{x} \neq \mathbf{x}_0 \tag{66}$$

$$\sup_{\|\mathbf{x}-\mathbf{x}_0\|>\epsilon} \langle \mathbf{R}(\mathbf{x}), V_\mathbf{x}(\mathbf{x})\rangle < 0 \quad \forall \epsilon > 0. \tag{67}$$

4) There exist constants $k_1, k_2 > 0$, such that

$$\begin{aligned}\|\mathbf{R}(\mathbf{x})\|^2 &+ \mathbb{E}[\|\boldsymbol{\Gamma}(t, \mathbf{x}, \omega)\|^2]\\ &\leq k_1(1+V(\mathbf{x})) - k_2\langle \mathbf{R}(\mathbf{x}), V_\mathbf{x}(\mathbf{x})\rangle.\end{aligned} \tag{68}$$

5) The weight sequence $\{\alpha(t)\}_{t\geq 0}$ satisfies the persistence condition D1 given by (62).

Then the Markov process $\{\mathbf{x}(t)\}_{t\geq 0}$ converges a.s. to $\mathbf{x}_0$.

Proof: The proof follows from [35, Theorem 4.4.4] and is omitted due to space constraints. ■

We now return to the proof of Theorem 3.

Proof (Proof of Theorem 3): We will show that, under the assumptions, the algorithm in (61) falls under the purview of Theorem 4. To this end, consider the filtration $\{\mathcal{F}_t\}_{t\geq 0}$, where

$$\mathcal{F}_t = \sigma(\mathbf{x}^j(0), \widetilde{\mathbf{S}}_\mathbf{P}(s), \widetilde{\mathbf{P}}(s),\ \widetilde{\mathbf{S}}_\mathbf{B}(s), \widetilde{\mathbf{B}}(s), \boldsymbol{\eta}^j(s) : 0 \leq s < t). \tag{69}$$

Define the vector $\mathbf{d}^*$ as

$$\mathbf{d}^* = (\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})^{-1}(\mathbf{B}+\mathbf{S}_\mathbf{B})\mathbf{u}^j. \tag{70}$$

Equation (61) can be written as

$$\begin{aligned}\mathbf{x}^j(t+1) = \mathbf{x}^j(t) - \alpha(t)[&(\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})(\mathbf{x}^j(t)-\mathbf{d}^*)\\ &- (\widetilde{\mathbf{S}}_\mathbf{P}(t)+\widetilde{\mathbf{P}}(t))\mathbf{x}^j(t)\\ &- (\widetilde{\mathbf{S}}_\mathbf{B}(t)+\widetilde{\mathbf{B}}(t))\mathbf{u}^j - \boldsymbol{\eta}^j(t)].\end{aligned} \tag{71}$$

In the notation of Theorem 4, (71) is given by

$$\mathbf{x}^j(t+1) = \mathbf{x}^j(t) + \alpha(t)\left[\mathbf{R}(\mathbf{x}^j(t)) + \boldsymbol{\Gamma}(t+1, \mathbf{x}^j(t), \omega)\right] \tag{72}$$

where

$$\mathbf{R}(\mathbf{x}) = -(\mathbf{I}-\mathbf{P}-\mathbf{S}_\mathbf{P})(\mathbf{x}^j(t)-\mathbf{d}^*) \tag{73}$$

and

$$\begin{aligned}\boldsymbol{\Gamma}(t+1, \mathbf{x}, \omega) = [&(\widetilde{\mathbf{S}}_\mathbf{P}(t)+\widetilde{\mathbf{P}}(t))\mathbf{x}^j(t)\\ &+ (\widetilde{\mathbf{S}}_\mathbf{B}(t)+\widetilde{\mathbf{B}}(t))\mathbf{u}^j + \boldsymbol{\eta}^j(t)].\end{aligned} \tag{74}$$

This definition satisfies assumptions 1) and 2) of Theorem 4. We now show the existence of a stochastic potential function $V(\cdot)$ satisfying the remaining assumptions of Theorem 4. Define

$$V(\mathbf{x}) = \|\mathbf{x}-\mathbf{d}^*\|^2. \tag{75}$$

[9] In the sequel, $\mathcal{B}^M$ denotes the Borel sigma algebra in $\mathbb{R}^{M\times 1}$. The space of twice continuously differentiable functions is denoted by $\mathcal{C}_2$, while $V_\mathbf{x}(\mathbf{x})$ denotes the gradient $(\partial V(\mathbf{x}))/(\partial \mathbf{x})$.

Clearly, $V(\mathbf{x}) \in \mathbb{C}_2$ with bounded second-order partial derivatives, with

$$V(\mathbf{d}^*) = 0, V(\mathbf{x}) > 0, \mathbf{x} \neq \mathbf{d}^*. \tag{76}$$

Also, we note that, for $\epsilon > 0$,

$$\begin{aligned}
&\sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} (\mathbf{R}(\mathbf{x}), V_{\mathbf{x}}(\mathbf{x})) \\
&\quad = \sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} -2(\mathbf{x}-\mathbf{d}^*)^T(\mathbf{I}-\mathbf{P}-\mathbf{S}_{\mathbf{P}})(\mathbf{x}-\mathbf{d}^*) \\
&\quad = \sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} [2(\mathbf{x}-\mathbf{d}^*)^T(\mathbf{P}+\mathbf{S}_{\mathbf{P}})(\mathbf{x}-\mathbf{d}^*) \\
&\qquad - 2\|\mathbf{x}-\mathbf{d}^*\|^2] \\
&\quad \leq \sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} [2|(\mathbf{x}-\mathbf{d}^*)^T(\mathbf{P}+\mathbf{S}_{\mathbf{P}})(\mathbf{x}-\mathbf{d}^*)| \\
&\qquad - 2\|\mathbf{x}-\mathbf{d}^*\|^2] \\
&\quad \leq \sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} [2\|\mathbf{x}-\mathbf{d}^*\|\rho(\mathbf{P}+\mathbf{S}_{\mathbf{P}})\|\mathbf{x}-\mathbf{d}^*\| \\
&\qquad - 2\|\mathbf{x}-\mathbf{d}^*\|^2] \\
&\quad = \sup_{\|\mathbf{x}-\mathbf{d}_*\|>\epsilon} -2(1-\rho(\mathbf{P}+\mathbf{S}_{\mathbf{P}}))\|\mathbf{x}-\mathbf{d}^*\|^2 \\
&\quad \leq -2\epsilon^2(1-\rho(\mathbf{P}+\mathbf{S}_{\mathbf{P}})) \\
&\quad < 0
\end{aligned} \tag{77}$$

where the last step follows from D2. Thus, assumption 3 in Theorem 4 is satisfied.

To verify 4, note that

$$\begin{aligned}
&\|\mathbf{R}(\mathbf{x})\|^2 \\
&\quad = (\mathbf{x}-\mathbf{d}^*)^T(\mathbf{I}-\mathbf{P}-\mathbf{S}_{\mathbf{P}})^T(\mathbf{I}-\mathbf{P}-\mathbf{S}_{\mathbf{P}})(\mathbf{x}-\mathbf{d}^*) \\
&\quad \leq \|(\mathbf{I}-\mathbf{P}-\mathbf{S}_{\mathbf{P}})^T(\mathbf{I}-\mathbf{P}-\mathbf{S}_{\mathbf{P}})\|\|\mathbf{x}-\mathbf{d}^*\|^2 \\
&\quad = k_1\|\mathbf{x}-\mathbf{d}^*\|^2 \\
&\quad = k_1 V(\mathbf{x})
\end{aligned} \tag{78}$$

where $k_1 > 0$ is a constant.

Finally, by assumptions C1–C4, we have

$$\begin{aligned}
&\mathbb{E}\|\mathbf{\Gamma}(t,\mathbf{x},\omega)\|^2 \\
&\quad = \mathbb{E}[(\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)+\widetilde{\mathbf{P}}(t-1))\mathbf{x}+(\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1) \\
&\qquad + \widetilde{\mathbf{B}}(t-1))\mathbf{u}^j+\boldsymbol{\eta}^j(t-1)]^T \\
&\qquad \times [(\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)+\widetilde{\mathbf{P}}(t-1))\mathbf{x}+(\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1) \\
&\qquad + \widetilde{\mathbf{B}}(t-1))\mathbf{u}^j+\boldsymbol{\eta}^j(t-1)] \\
&\quad = \mathbf{x}^T\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)+\widetilde{\mathbf{P}}^T(t-1)\widetilde{\mathbf{P}}(t-1)]\mathbf{x} \\
&\qquad + \mathbf{u}^{jT}\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{B}}^T(t-1)\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1) \\
&\qquad + \widetilde{\mathbf{B}}^T(t-1)\widetilde{\mathbf{B}}(t-1)]\mathbf{u}^j+\mathbb{E}[\|\boldsymbol{\eta}^j(t-1)\|^2] \\
&\qquad + 2\mathbf{x}^T\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1)]\mathbf{u}^j \\
&\quad \leq \mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)\|^2+\|\widetilde{\mathbf{P}}(t-1)\|^2]\|\mathbf{x}\|^2 \\
&\qquad + \mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1)\|^2+\|\widetilde{\mathbf{B}}(t-1)\|^2]\|\mathbf{u}^j\|^2 \\
&\qquad + \mathbb{E}[\|\boldsymbol{\eta}^j(t-1)\|^2]+2\mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)\|^2]^{1/2} \\
&\qquad \times \mathbb{E}[\|\widetilde{\mathbf{S}}_{\mathbf{B}}(t-1)\|^2]^{1/2}\|\mathbf{x}\|\|\mathbf{u}^j\| \\
&\quad \leq (k_{\mathbf{P}}+\widetilde{k}_{\mathbf{P}})\|\mathbf{x}\|^2+(k_{\mathbf{B}}+\widetilde{k}_{\mathbf{B}})\|\mathbf{u}^j\|^2 \\
&\qquad + k_\eta + k_{\mathbf{P}}k_{\mathbf{B}}\|\mathbf{x}\|\|\mathbf{u}^j\|.
\end{aligned} \tag{79}$$

The cross terms dropped in the second step of (79) have zero mean by the independence assumption C4. For example, consider the term $\mathbb{E}[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{P}}(t-1)]$. It follows from (51) and (56) that the lnth entry of the matrix $\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{P}}(t-1)$ is given by

$$\begin{aligned}
&\left[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{P}}(t-1)\right]_{ln} \\
&\quad = \sum_r \left[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\right]_{lr}[\widetilde{\mathbf{P}}(t-1)]_{rn} \\
&\quad = \sum_r [\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\widehat{\mathbf{P}}(t-1)]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right) \\
&\quad = \sum_r [\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\mathbf{P}]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right) \\
&\qquad + \sum_r [\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\mathbf{S}_{\mathbf{P}}]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right) \\
&\qquad + \sum_r [\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right).
\end{aligned} \tag{80}$$

From the independence and zero-mean assumptions, we have the following, $\forall r$:

$$\begin{aligned}
&\mathbb{E}\left[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\mathbf{P}]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = [\mathbf{P}]_{rn}\mathbb{E}[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}]\mathbb{E}\left[\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = 0 \\
&\mathbb{E}\left[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\mathbf{S}_{\mathbf{P}}]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = [\mathbf{S}_{\mathbf{P}}]_{rn}\mathbb{E}[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}]\mathbb{E}\left[\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = 0 \\
&\mathbb{E}\left[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rn}\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = \mathbb{E}\left[[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rl}[\widetilde{\mathbf{S}}_{\mathbf{P}}(t-1)]_{rn}\right]\mathbb{E}\left[\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] \\
&\quad = 0
\end{aligned}$$

where we have repeatedly used the fact that

$$\mathbb{E}\left[\left(\frac{e_{rn}(t)}{q_{rn}}-1\right)\right] = 0. \tag{81}$$

From (80)–(81) it is then clear that

$$\mathbb{E}\left[\widetilde{\mathbf{S}}_{\mathbf{P}}^T(t-1)\widetilde{\mathbf{P}}(t-1)\right] = 0. \tag{82}$$

In a similar way, it can be shown that the other dropped crossed terms in (79) are zero mean.

We note that there exist constants, $k_3, k_4, k_5, k_6 > 0$, such that

$$\begin{aligned}
\|\mathbf{x}\|^2 &\leq k_3\|\mathbf{x}-\mathbf{d}^*\|^2 + k_4 \\
\|\mathbf{x}\| &\leq k_5\|\mathbf{x}-\mathbf{d}^*\|^2 + k_6.
\end{aligned} \tag{83}$$

Hence, from (79) and (83), we have

$$\begin{aligned}\mathbb{E}\|\boldsymbol{\Gamma}(t,\mathbf{x},\omega)\|^2 &\leq k_7\|\mathbf{x}-\mathbf{d}^*\|^2+k_8,\\ &\leq k_9(1+V(\mathbf{x}))\end{aligned} \tag{84}$$

where $k_7, k_8 > 0$ and $k_9 = \max(k_7, k_8)$. Combining (78) and (84), we note that assumption 4 in Theorem 4 is satisfied as

$$(\mathbf{R}(\mathbf{x}), V_{\mathbf{x}}(\mathbf{x})) \leq 0 \quad \forall \mathbf{x}. \tag{85}$$

Hence, all the conditions of Theorem 4 are satisfied, and we conclude that

$$\mathbf{P}[\lim_{t\to\infty}\mathbf{x}^j(t) = (\mathbf{I}-\mathbf{P}-\mathbf{S_P})^{-1}(\mathbf{B}+\mathbf{S_B})\mathbf{u}^j] = 1. \tag{86}$$

Since (86) holds for all j, and j takes only a finite number of values $(1 \leq j \leq m)$, we have

$$\mathbf{P}[\lim_{t\to\infty}\mathbf{x}^j(t) = (\mathbf{I}-\mathbf{P}-\mathbf{S_P})^{-1}(\mathbf{B}+\mathbf{S_B})\mathbf{u}^j, \quad \forall j] = 1. \tag{87}$$

■

We now interpret Theorem 3. Referring to the partitioning of the $\mathbf{C}(t)$ matrix in (47), we have

$$\mathbf{C}(t) = \begin{bmatrix}\mathbf{U}\\ \mathbf{X}(t)\end{bmatrix} \tag{88}$$

where each row of $\mathbf{X}(t)$ corresponds to an estimated sensor location at time t. Theorem 3 then states that, starting with any initial guess, $\mathbf{X}(0) \in \mathbb{R}^{M\times m}$, of the unknown sensor locations, the state sequence $\{\mathbf{X}(t)\}_{t\geq 0}$ generated by the DLRE algorithm converges a.s., i.e.,

$$\mathbf{P}[\lim_{t\to\infty}\mathbf{X}(t) = (\mathbf{I}-\mathbf{P}-\mathbf{S_P})^{-1}(\mathbf{B}+\mathbf{S_B})\mathbf{U}] = 1. \tag{89}$$

However, it follows from Lemma 4, that the exact locations of the unknown sensors are given by

$$\mathbf{X}^* = (\mathbf{I}-\mathbf{P})^{-1}\mathbf{B}\mathbf{U}. \tag{90}$$

Thus, the steady-state estimate given by the DLRE algorithm is not exact, and, to characterize its performance, we introduce the following notion of localization error e_l as

$$e_l = \|(\mathbf{I}-\mathbf{P}-\mathbf{S_P})^{-1}(\mathbf{B}+\mathbf{S_B})\mathbf{U}-(\mathbf{I}-\mathbf{P})^{-1}\mathbf{B}\mathbf{U}\|. \tag{91}$$

We note that e_l is only a function of $\mathbf{S_P}, \mathbf{S_B}$, the nonzero biases in the system matrix computations, resulting from noisy internode distance measurements (see Section V). We note that the DLRE algorithm is robust to random link failures and additive channel noise in internode communication. In fact, it is also robust to the zero-mean random errors in the system matrix computations, and only affected by the fixed nonzero biases. Note that $e_l = 0$ for $\mathbf{S_P} = \mathbf{S_B} = 0$. Clearly, if we assume sufficient accuracy in the internode distance computation process, so that the biases $\mathbf{S_P}, \mathbf{S_B}$ are small, the steady-state error e_l will also be negligible, even in a random sensing environment. These are illustrated by numerical studies provided in the next section.

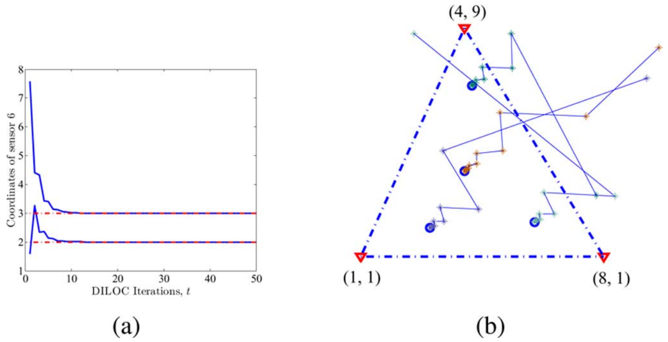

Fig. 3. Deterministic environment: (a) Estimated coordinates of sensor 6 in Section II-C as a function of DILOC iterations. (b) Trajectories of the sensors' estimates obtained by DILOC.

VII. Numerical Studies

We carry out two studies of the localization algorithm in $m = 2$-dimensional Euclidean space. First, we study DILOC (the deterministic case), and second, we present DLRE in random environments.

DILOC: We consider the example presented in Section II-C, which has $m+1 = 3$ anchors and $M = 4$ sensors with no noise in the system. DILOC, as given in (14), is implemented, where Fig. 3(a) shows the estimated coordinates of sensor 6, and Fig. 3(b) shows the trajectories of the estimated coordinates for all the sensors with random initial conditions. Both figures show fast convergence to the exact sensor locations, which should be the case because of the geometric convergence rate.

We further consider a network of $N = 500$ nodes shown in Fig. 4(a) after triangulation. The communication radius is increased until each sensor triangulates. DILOC is implemented with zero initial conditions and Fig. 4(b) shows the estimated coordinates of two arbitrary sensors; this illustrates that geometric convergence of DILOC estimates to the exact sensor locations. Fig. 4(c) shows a typical histogram of the internode distances (normalized by the mean of all anchor-anchor distances) over which the DILOC communications are implemented. It can be verified that the ninety-fifth percentile of the internode distances are within 10% of the mean anchor-anchor distance.

DLRE: We now consider the DLRE. To simulate the random scenario, we assume that the communication links are active 90% of the time, i.e., the probability, $q_{ln} = 0.9 \quad \forall l, n,$s.t. $l \in \Omega\, n \in \Theta_l$, as discussed in C2, and there is an additive communication noise that is Gaussian i.i.d. with zero-mean and variance 1 for each link. We further assume that the perturbation matrices, $\widetilde{\mathbf{S}}_{\mathbf{P}}(t), \widetilde{\mathbf{S}}_{\mathbf{B}}(t)$ are zero-mean Gaussian i.i.d. with variance 0.1 and $\mathbf{S_P} = 0, \mathbf{S_B} = 0$; see (49) and (51). Recall that the elements of $\mathbf{B}$ and $\mathbf{P}$ lie in the unit interval $[0, 1]$ so the variance chosen for the simulation is a small signal perturbation of these elements.

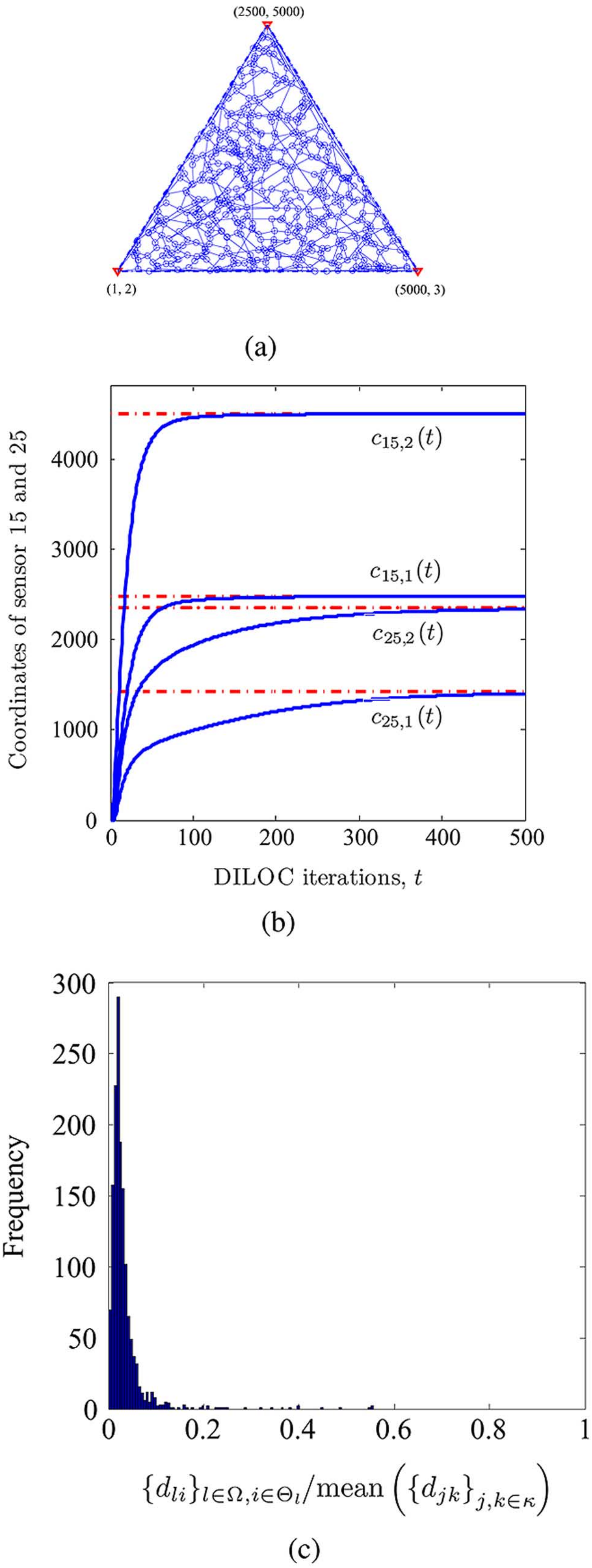

Fig. 4. Deterministic environment: (a) An $N = 500$ node network and the respective triangulation sets. (b) Estimated coordinates of two arbitrarily chosen sensors as a function of DILOC iterations. (c) Histogram of normalized internode distances over which the DILOC communications are implemented.

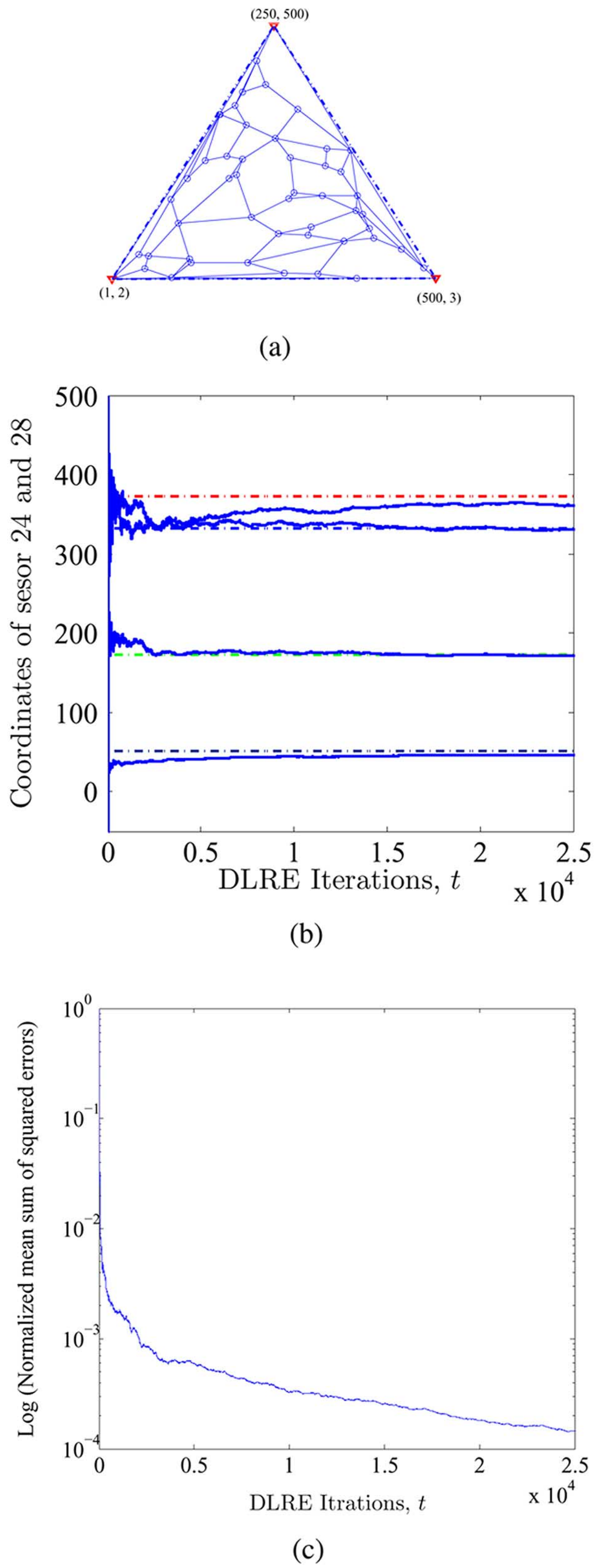

Fig. 5. Random environments (noisy distances, communication noise, link failures): (a) $N = 50$ node network and the respective triangulation sets. (b) DLRE estimates for two arbitrarily chosen sensors. (c) Log of the normalized MSE_t.

DLRE is implemented for the above setup with zero initial conditions on a network of $N = 50$ nodes shown in Fig. 5(a). We use a decreasing weight sequence $\alpha(t) = 1/t^{0.55}$, and the estimated coordinates for two arbitrarily chosen sensors are shown in

Fig. 5(b). The estimates converge to the exact sensor locations. To study the error decay, we consider the following quantity:

$$\mathrm{MSE}_t = \frac{1}{M}\sum_{l=1}^{M}\sum_{i=1}^{m}(\mathbf{c}_{l,i}(t) - \mathbf{c}^*_{l,i})^2 \tag{92}$$

and plot $\log[\mathrm{MSE}_t/\max_t(\mathrm{MSE}_t)]$ in Fig. 5(c). We use the log-scale in Fig. 5(c) so the error plot is visible. Otherwise, due to the fast convergence speed, the plot looks almost like a vertical line. The fast convergence rate of the DLRE estimates can be verified since the estimates become very close to the exact locations in a very few iterations.

VIII. Conclusion

The paper presents an algorithm for distributed iterative sensor localization in m-dimensional Euclidean space $\mathbb{R}^m (m \geq 1)$ that finds the location coordinates of the sensors in a sensor network with only local communication. The algorithm uses the minimal number $m+1$ of anchors (network nodes with known location) to localize an arbitrary number M of sensors that lie in the convex hull of these $m+1$ anchors. In the deterministic case, i.e., when there is no noise in the system, we show that our distributed algorithms, DILOC and DILOC-REL, lead to convergence to the exact sensor locations. For the random environment scenario, where internode communication links may fail randomly, transmitted data is distorted by noise, and internode distance information is imprecise, we show that our modified algorithm DLRE leads to almost sure convergence of the iterative location estimates. In this case, we explicitly characterize the resulting error between the exact sensor locations and the converged estimates. Numerical simulations illustrate the behavior of the algorithms under different field conditions.

As long as the weight sequence $\alpha(t)$ satisfies the persistence conditions D1, the DLRE converges a.s. However, the second-order characteristics of convergence like rate, smoothness of the trajectory, etc., depend on the particular choice of $\alpha(t)$, which is studied in the context of distributed stochastic consensus averaging in [33]. In the future, we would like to pursue this analysis in the context of DLRE. Natural extensions to the results in this paper, i.e., with more than $m+1$ anchors, dynamic network topology and more than $m+1$ neighbors in the triangulation set Θ_l are studied in [36].

Appendix I
Convex Hull Inclusion Test

We now give an algorithm that tests if a given sensor, $l \in \mathbb{R}^m$, lies in the convex hull of $m+1$ nodes in a set, κ, using only the mutual distance information among these $m+2$ nodes $(\kappa \cup \{l\})$. Let κ denote the set of $m+1$ nodes, and let $\mathcal{C}(\kappa)$ denote the convex hull formed by the nodes in κ. Clearly, if $l \in \mathcal{C}(\kappa)$, then the convex hull formed by the nodes in κ is the same as the convex hull formed by the nodes in $\kappa \cup \{l\}$, i.e.,

$$\mathcal{C}(\kappa) = \mathcal{C}(\kappa \cup \{l\}), \quad \text{if } l \in \mathcal{C}(\kappa). \tag{93}$$

With the above equation, we can see that, if $l \in \mathcal{C}(\kappa)$, then the generalized volumes of the two convex sets $\mathcal{C}(\kappa)$ and $\mathcal{C}(\kappa \cup \{l\})$ should be equal. Let A_κ denote the generalized volume of $\mathcal{C}(\kappa)$, and let $A_{\kappa\cup\{l\}}$ denote the generalized volume of $\mathcal{C}(\kappa \cup \{l\})$, we have

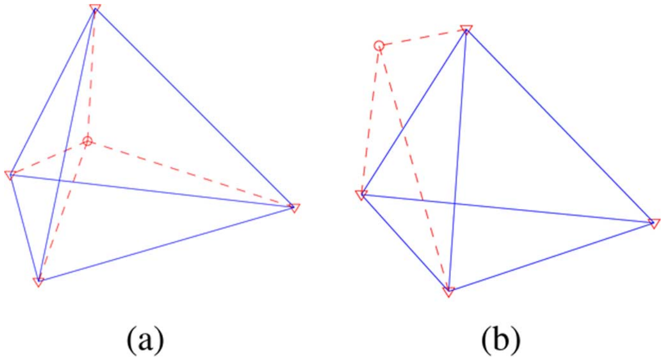

Fig. 6. Convex Hull Inclusion Test ($m=3$): The sensor l is shown by a "o", whereas the anchors in κ are shown by "∇". (a) $l \in \mathcal{C}(\kappa) \Rightarrow A_\kappa = A_{\kappa\cup\{l\}}$ and (b) $l \notin \mathcal{C}(\kappa) \Rightarrow A_\kappa < A_{\kappa\cup\{l\}}$.

$$A_\kappa = A_{\kappa\cup\{l\}} = \sum_{k\in\kappa} A_{\kappa\cup\{l\}\setminus\{k\}}, \quad \text{if } l \in \mathcal{C}(\kappa). \tag{94}$$

Hence, the test becomes

$$l \in \mathcal{C}(\kappa), \quad \text{if } \sum_{k\in\kappa} A_{\kappa\cup\{l\}\setminus\{k\}} = A_\kappa \tag{95}$$

$$l \notin \mathcal{C}(\kappa), \quad \text{if } \sum_{k\in\kappa} A_{\kappa\cup\{l\}\setminus\{k\}} > A_\kappa. \tag{96}$$

This is also shown in Fig. 6. The above inclusion test is based entirely on the generalized volumes, which can be calculated using only the distance information in the Cayley–Menger determinants.

Appendix II
Cayley–Menger Determinant

The Cayley–Menger determinant [37] is the determinant of an $m+2 \times m+2$ (symmetric) matrix that relates to the generalized volume A_{Θ_l} of the convex hull $\mathcal{C}(\Theta_l)$ of the $m+1$ points in $\mathbb{R}^m$ through an integer sequence s_{m+1}. Let $\mathbf{1}_{m+1}$ denote a column vector of $m+1$ 1 s, the Cayley–Menger determinant is given by

$$A^2_{\Theta_l} = \frac{1}{s_{m+1}}\begin{vmatrix} 0 & \mathbf{1}^T_{m+1} \\ \mathbf{1}_{m+1} & \mathbf{Y} \end{vmatrix} \tag{97}$$

where $\mathbf{Y} = \{d^2_{lj}\}, l, j \in \Theta_l$ is the matrix of squared distances d_{lj} among the $m+1$ points in Θ_l and

$$s_m = \frac{2^m(m!)^2}{(-1)^{m+1}}, \quad m = \{0, 1, 2, \ldots\}. \tag{98}$$

Appendix III
Important Results

Lemma 5: If the matrix $\mathbf{P}$ corresponds to the transition probability matrix associated to the transient states of an absorbing Markov chain, then

$$\lim_{t\to\infty} \mathbf{P}^{t+1} = \mathbf{0}. \tag{99}$$

Proof: For such a matrix $\mathbf{P}$, we have

$$\rho(\mathbf{P}) < 1 \tag{100}$$

from Lemma 8.3.20 and Theorem 8.3.21 in [38], where $\rho(\cdot)$ denotes the spectral radius of a matrix and (99) follows from (100). ■

Lemma 6: If the matrix $\mathbf{P}$ corresponds to the transition probability matrix associated to the transient states of an absorbing Markov chain, then

$$\lim_{t\rightarrow\infty}\sum_{k=0}^{t+1}\mathbf{P}^k = (\mathbf{I}-\mathbf{P})^{-1}. \tag{101}$$

Proof: The proof follows from Lemma 5 and Lemma 6.2.1 in [38]. ■

REFERENCES

[1] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[2] G. Springer, *Introduction to Riemann Surfaces*. Reading, MA: Addison-Wesley, 1957.

[3] J. G. Hocking and G. S. Young, *Topology*. Reading, MA: Addison-Wesley, 1961.

[4] R. L. Moses, D. Krishnamurthy, and R. Patterson, "A self-localization method for wireless sensor networks," *EURASIP J. Appl. Signal Process.*, no. 4, pp. 348–358, Mar. 2003.

[5] N. Patwari, A. O. Hero, III, M. Perkins, N. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[6] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Networking Computing*, Annapolis, MD, Jun. 2003, pp. 201–212.

[7] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 2640–2651.

[8] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 93–101, Feb. 2005.

[9] M. Cao, B. D. O. Anderson, and A. S. Morse, "Localization with imprecise distance information in sensor networks," in *Proc. 44th IEEE Conf. Decision Control and European Control Conf. 2005 (CDC-ECC)*, Sevilla, Spain, Dec. 2005, pp. 2829–2834.

[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec. 2000.

[11] N. Patwari and A. O. Hero, III, "Manifold learning algorithms for localization in wireless sensor networks," in *Proc. 29th IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, QC, Canada, Mar. 2004, vol. 3, pp. 857–860.

[12] D. Niculescu and B. Nath, "Ad-hoc positioning system (aps)," in *Proc. IEEE GLOBECOM*, San Antonio, TX, Apr. 2001, pp. 2926–2931.

[13] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad hoc networks of sensors," in *Proc. IEEE MOBICOM*, Rome, Italy, Jul. 2001, pp. 166–179.

[14] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems," in *Proc. Int. Workshop Sensor Networks Applications*, Atlanta, GA, Sep. 2002, pp. 112–121.

[15] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proc. 2nd Int. Workshop Information Processing Sensor Networks*, Palo Alto, CA, Apr. 2003, pp. 333–348.

[16] J. J. Caffery, *Wireless Location in CDMA Cellular Radio Systems*. Norwell, MA: Kluwer Academic, 1999.

[17] J. A. Costa, N. Patwari, and A. O. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sens. Netw.*, vol. 2, no. 1, pp. 39–64, 2006.

[18] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Proc. IEEE Int. Conf. Network Protocols*, Riverside, CA, Nov. 2001, pp. 35–41.

[19] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad hoc wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Salt Lake City, UT, May 2001, pp. 2037–2040.

[20] S. Čapkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad hoc networks," in *Proc. 34th IEEE Hawaii Int. Conf. System Sciences, Cluster Computing*, Wailea Maui, HI, Jan. 2001, pp. 3481–3490.

[21] A. T. Ihler, J. W. Fisher, III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-calibration in sensor networks," in *Proc. 29th IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, QC, Canada, May 2004, pp. 265–268.

[22] L. Hu and D. Evans, "Localization for mobile sensor networks," in *IEEE MOBICOM*, Philadelphia, PA, Sep. 2004, pp. 45–57.

[23] M. Coates, "Distributed particle filters for sensor networks," in *Proc. IEEE Information Processing Sensor Networks (IPSN)*, Berkeley, CA, Apr. 2004, pp. 99–107.

[24] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, Mar. 2002.

[25] J. C. Gower, "Euclidean distance geometry," *Math. Scient.*, vol. 7, pp. 1–14, 1982.

[26] J. C. Gower, "Properties of Euclidean and non-Euclidean distance matrices," *Linear Algebra Appl.*, vol. 67, pp. 81–97, Jun. 1985.

[27] N. Patwari, "Location estimation in sensor networks," Ph.D. dissertation, Univ. of Michigan, Ann Arbor, MI, 2005.

[28] U. Khan, S. Kar, and J. M. F. Moura, "Distributed algorithms in sensor networks," in *Handbook on Sensor and Array Processing*, S. Haykin and K. J. Ray Liu, Eds. New York: Wily-Interscience, 2009, to be published.

[29] U. A. Khan, S. Kar, and J. M. F. Moura, "Higher dimensional consensus algorithms in sensor networks," in *Proc. 34th IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Taipei, Taiwan, Apr. 2009, submitted for publication.

[30] P. Hall, *Introduction to the Theory of Coverage Processes*. Chichester, U.K.: Wiley, 1988.

[31] Y. Sung, L. Tong, and A. Swami, "Asymptotic locally optimal detector for large-scale sensor networks under the Poisson regime," *IEEE Trans. Signal Process.*, vol. 53, no. 6, pp. 2005–2017, Jun. 2005.

[32] C. M. Grinstead and J. L. Snell, *Introduction to Probability*. Providence, RI: Amer. Math. Soc., 1997.

[33] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Process.* vol. 57, no. 1, pp. 355–369, Jan. 2009.

[34] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data," Dec. 2007 [Online]. Available: http://aps.arxiv.org/abs/0712.1609, submitted for publication

[35] M. B. Nevel'son and R. Z. Has'minskii, *Stochastic Approximation and Recursive Estimation*. Providence, RI: Amer. Math. Soc., 1973.

[36] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in Euclidean spaces: Dynamic environments," presented at the 46th Allerton Conf. Communication, Control, Computing, Monticello, IL, Sep. 2008.

[37] M. J. Sippl and H. A. Scheraga, "Cayley–Menger coordinates," *Proc. Nat. Acad. Sciences U.S.A.*, vol. 83, no. 8, pp. 2283–2287, Apr. 1986.

[38] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*. New York: Academic, 1970.

Usman A. Khan (S'99) received the B.S. degree (with hons.) in electrical engineering from the University of Engineering and Technology, Lahore-Pakistan, in 2002, and the M.S. degree in electrical and computer engineering from the University of Wisconsin-Madison in 2004. Currently, he is working towards the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, under the supervision of Prof. J. M. F. Moura.

He worked as a Researcher in the National Magnetic Resonance Facility at Madison (NMRFAM) from 2003 to 2005, where he worked on shape characterization of protein molecules. He was a Research Assistant in the Computer Science Department at the University of Wisconsin-Madison from 2004 to 2005, where he worked on statistical estimation techniques for NMR signals. He worked as an intern in AKAMAI Technologies during summer 2007. His research interests include statistical signal processing for large-scale dynamical systems, networked control and estimation, and distributed linear/nonlinear iterative algorithms.

Mr. Khan holds an ECE fellowship from Carnegie Mellon University.

Soummya Kar (S'02) received the Bachelor of Technology (BTech) degree from the Indian Institute of Technology, Kharagpur, India, in 2005. He is currently working towards the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA.

His research interests include distributed computations in sensor networks, stochastic approximations, stochastic networks, and stochastic filtering in the context of sensor networks.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the Engenheiro Electrotécnico degree from the Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

He is a Professor of electrical and computer engineering and, by courtesy, of biomedical engineering, at Carnegie Mellon University (CMU). He was on the faculty at IST, has held visiting faculty appointments at MIT, and was a visiting Research Scholar at the University of Southern California, Los Angeles. He is a founding co-director of the Center for Sensed Critical Infrastructures Research (CenSCIR) and manages a large education and research program between CMU and Portugal (www.icti.cmu.edu). His research interests include statistical and algebraic signal processing, image, bioimaging, and video processing, and digital communications. He has published over 340 technical journal and conference papers, is the co-editor of two books, holds seven patents, and has given numerous invited seminars at international conferences, U.S. and European universities, and industrial and government Laboratories.

Dr. Moura is the President (2008–2009) of the IEEE Signal Processing Society (SPS). He was Editor-in-Chief for the IEEE TRANSACTIONS IN SIGNAL PROCESSING, interim Editor-in-Chief for the IEEE SIGNAL PROCESSING LETTERS, and was on the Editorial Board of several journals, including the IEEE PROCEEDINGS, the IEEE SIGNAL PROCESSING MAGAZINE, and the ACM *Transactions on Sensor Networks*. He was on the steering and technical committees of several Conferences. He is a Fellow of the IEEE, a Fellow of the American Association for the Advancement of Science (AAAS), and a corresponding member of the Academy of Sciences of Portugal (Section of Sciences). He was awarded the 2003 IEEE Signal Processing Society Meritorious Service Award and in 2000 the IEEE Millennium Medal. In 2007, he received the CMU's College of Engineering Outstanding Research Award, and, in 2008, the Philip L. Dowd Fellowship Award. He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.