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

Usman A. Khan†, Soummya Kar†, and José M. F. Moura†

Department of Electrical and Computer Engineering
Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213
{ukhan, moura}@ece.cmu.edu, soummyak@andrew.cmu.edu
Ph: (412)268-7103 Fax: (412)268-3890

Abstract

The paper develops DILOC, a distributive, iterative algorithm that locates $M$ sensors in $\mathbb{R}^m, m \geq 1$, with respect to a minimal number of $m+1$ anchors with known locations. The sensors exchange data with their neighbors only; no centralized data processing or communication occurs, nor is there centralized knowledge about the sensors’ locations. DILOC uses the barycentric coordinates of a sensor with respect to its neighbors that are computed using the Cayley-Menger determinants. These are the determinants of matrices of inter-sensor distances. We show convergence of DILOC by associating with it an absorbing Markov chain whose absorbing states are the anchors. We introduce a stochastic approximation version extending DILOC to random environments when the knowledge about the intercommunications among sensors and the inter-sensor distances are noisy, and the communication links among neighbors fail at random times. We show a.s. convergence of the modified DILOC and characterize the error between the final estimates and the true values of the sensors’ locations. Numerical studies illustrate DILOC under a variety of deterministic and random operating conditions.

Keywords: Distributed iterative sensor localization; sensor networks; Cayley-Menger determinant; barycentric coordinates; absorbing Markov chain; stochastic approximation.

†All authors contributed equally to the paper. This work was partially supported 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.
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. To be more specific, we are motivated by applications where \( N = M + m + 1 \) sensors in \( \mathbb{R}^m \) (for example, \( m = 2 \) corresponds to sensors lying on a plane, while for \( m = 3 \) the sensors are in three dimensional Euclidean space) are deployed in a large geographical region. We assume that the deployment region lies in the convex hull of a small, in fact minimal, number of \( m + 1 \) anchors, \( (m + 1) \ll M \). The anchors know their locations. In such situations, the large geographical distances to the anchors makes it highly impractical for the \( M \) non-anchor sensors to communicate directly with the anchors. Further, to compute the locations of the non-anchor sensors at a central station is not feasible when \( M \) is large, as it requires large communication effort, expensive large-scale computation, and adds latency and bottlenecks to the network operation. These networks call for efficient distributed algorithms where each sensor communicates directly only with a few neighboring nodes (either sensors or anchors) and a low order computation is performed locally at the sensor and at each iteration of the algorithm, for example, see [1]. In this paper, we present a 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. This state updating is a convex combination of the states of the neighboring nodes. The coefficients of the convex combination are the barycentric coordinates [2], [3] and are determined from the mutual inter-sensor distances among the sensors using the Cayley-Menger determinants. At each sensor \( l \) its neighborhood set contains \( m + 1 \) sensors, for example, its closest \( m + 1 \) sensors, such that sensor \( l \) lies in the convex hull of these \( m + 1 \) neighbors. These neighbors may or may not include the anchors.

DILOC is distributed and iterative; each sensor 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 inter-sensor 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 sensors fail at random times. This is mainly motivated by wireless digital communication, where packets may get dropped randomly at each iteration, particularly, if the sensors are power limited or there are bandwidth or rate communication constraints in
the network; and (iii) the communication among two sensors, when their communication link is active, is corrupted by noise.

Although a sensor can only communicate directly with its neighbors (e.g., sensors 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 sensors. In a random environment, inter-sensor 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 as Distributed Localization in Random Environments (DLRE). 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], [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], [13], [14], [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 that have better accuracy. This allows us to employ a minimal number of anchors.

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], [19], [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. Non-parametric 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], [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 our algorithm is distributed. In particular, it does not require a centralized location to perform the computations.

In comparison with these references, our algorithm, DILOC, is equivalent to solving in a distributed and iterative fashion 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 (distributed iterative localization algorithm) in \( m \)-dimension Euclidean space, \( \mathbb{R}^m \) \((m \geq 1)\), and introduce the necessary 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 reference, 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 straight forwardly 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. Preliminaries and Notation

Recall that the sensors are in \( \mathbb{R}^m \). Let \( \Theta \) be the set of sensors or nodes in the network decomposed as

\[
\Theta = \kappa \cup \Omega,
\]

where \( \kappa \) is the set of anchors, i.e., the sensors whose locations are known, and \( \Omega \) 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 \( \Psi \) of sensors, we denote its convex hull by \( C(\Psi) \). For example, if \( \Psi \) is a set of three non-coplanar sensors in a plane, then \( C(\Psi) \) is a triangle. We now define a few additional sets needed.

Let \( d_{lk} \) be the Euclidean distance between two sensors \( l, k \in \Theta \). We associate with the sensor \( l \in \Omega \), a positive real number, \( r_l \), and two sets \( K(l, r_l) \) and \( \Theta_l(r_l) \):

\[
K(l, r_l) = \{ k \in \Theta : d_{lk} < r_l \},
\]

\[
\Theta_l(r_l) \subseteq K(l, r_l), \ l \notin \Theta_l(r_l), \ l \in C(\Theta_l(r_l)), \ |\Theta_l(r_l)| = m + 1, \ A_{\Theta_l(r_l)} \neq 0,
\]

where \( A_{\Theta_l(r_l)} \) is the generalized volume (area in \( m = 2 \)-d, volume in \( m = 3 \), and their generalization in higher dimensions) of \( C(\Theta_l(r_l)) \). The set \( K(l, r_l) \) groups the neighboring sensors of \( l \) within a radius \( r_l \) and, by (3), \( \Theta_l(r_l) \), which we will often represent simply as \( \Theta_l \), assuming \( r_l \) is understood from the context, is a subset of \( m + 1 \) sensors such that sensor \( l \) lies in its convex hull but is not one of its elements. In appendix I, we provide 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, \( K(l, r_l) \), of its neighbors. Finding such a set \( \Theta_l \) is an important step in DILOC and we refer to it as triangulation and \( \Theta_l \) is referred to as a triangulation set.

Let \( c_l \) be the \( m \)-dimensional coordinates for a node, \( l \in \Theta \), with respect to a global coordinate system, written as the \( m \)-dimensional row vector,

\[
c_l = [c_{l,1}, c_{l,2}, \ldots, c_{l,m}].
\]

The true (possibly unknown) location of sensor \( l \) is represented by \( c_l^* \). Because the distributed localization algorithm DILOC is iterative, \( c_l(t) \) will represent the location vector, or state, for sensor \( l \) at iteration \( t \).

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

\[
a_{lk} = \frac{A_{\{l\} \cup \Theta_l \setminus \{k\}}}{A_{\Theta_l}},
\]

\(^1\)The convex hull, \( C(\Psi) \), of a set of points in \( \Psi \) is the minimal convex set containing \( \Psi \).
with $A_{\Theta_l} \neq 0$, where \( \setminus \) denotes the set difference, $A_{\{l\} \cup \Theta_l \setminus \{k\}}$ is the generalized volume of the set $\{l\} \cup \Theta_l \setminus \{k\}$, i.e., the set $\Theta_l$ with sensor $l$ added and node $k$ removed. The barycentric coordinates can be computed from the inter-sensor distances $d_{lk}$ using the Cayley-Menger determinants as shown in appendix II. From (5), and the facts that the generalized volumes are non-negative and

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

(6)

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

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

(7)

B. Distributed iterative localization algorithm.

Before presenting DILOC, we state explicitly state our assumptions.

(B0) Nondegeneracy. The generalized volume for $\kappa$, $A_\kappa \neq 0$.\(^2\)

(B1) Anchor nodes. The anchors’ locations are known, i.e., their state remains constant $c_q(t) = c^*_q \quad q \in \kappa, \quad t \geq 0$. \(^8\)

(B2) Convexity. All the sensors lie inside the convex hull of the anchors $C(\Omega) \subset C(\kappa)$. \(^9\)

From (B2), the next Lemma follows easily.

Lemma 1 (Triangulation) For every $l \in \Omega$, there exist $r_l > 0$ and $\Theta_l(r_l)$ with $|\Theta_l(r_l)| = m + 1$ satisfying the properties in (3).

**Proof:** Clearly, by (B2), $\kappa$ satisfies (3) and the diameter of the network, $\max_{l,k} d_{lk}, (l \in \Omega, k \in \kappa)$, could be taken as $r_l$.

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-E discusses the probability of finding one such $\Theta_l$ with $r_l \ll \max_{l,k} d_{lk}, (l \in \Omega, k \in \kappa)$.

As a note, it is easily verified that every pair of sensors in $C(\Theta_l(r_l))$ is within a distance $R_l = 2r_l$. Think of $m = 2$, then $|\Theta_l(r_l)| = 3$, $C(\Theta_l(r_l))$ is a triangle and if $r_l$ is the maximum distance of any interior point $l$ to the vertices of the triangle, the maximum distance between the 4 points (triangle vertices and $l$) is $2r_l$. We complete stating the assumptions underlying DILOC.

\(^2\)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 a network lie on a plane in $\mathbb{R}^3$, the localization problem can be thought of as localization in $\mathbb{R}^2$. 

August 7, 2008 DRAFT
(B3) inter-sensor distances. For \( l \in \Omega \), there exists at least an \( r_l > 0 \) and \( \Theta_l (r_l) \subset K (l, r_l) \), satisfying (3), such that \( l \) has a communication link for every \( k \in \Theta_l (r_l) \) and knows the mutual distances among all the nodes in \( \{ l \} \cup \Theta_l (r_l) \).

We can now present DILOC. There are two steps: a set-up phase and then DILOC proper. We discuss each separately.

**DILOC set-up: Triangulation.** In the set-up phase, each sensor \( l \) triangulates itself, so that by the end of this phase we have paired every \( l \in \Omega \) with its corresponding \( m + 1 \) neighbors in \( \Theta_l \). Since triangulation should be with a small \( r_l \), the following is a practical protocol for the set-up phase. A sensor starts with a small communication radius, \( r_l \), and chooses arbitrarily \( m + 1 \) sensors within \( r_l \) and tests if it lies in the convex hull of these sensors using the procedure described in Appendix I. The sensor attempts this with all collections of \( m + 1 \) sensors within \( r_l \).

If all attempts fail, the sensor adaptively increases, in small increments, its communication radius, \( r_l \), and repeats the process. By (B2) and (9), success is eventually achieved and each sensor is triangulated by finding \( \Theta_l \) with properties (3) and (B3). If the sensors have directionality a much simpler algorithm based on Lemma 2 below, see also discussion following the Lemma, triangulates the sensor with high probability of success in one shot. To assess the practical implications required by DILOC’s set-up phase, Subsection II-E 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 \( \Theta_l \) in a given radius, \( r_l \).

**DILOC iterations: state updating.** Once the set-up 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 \( \Theta_l \). The anchors do not update their state, since they know their locations. The updating is explicitly given by

\[
c_l (t + 1) = \begin{cases} c_l (t), & l \in \kappa, \\ \sum_{k \in \Theta_l} a_{lk} c_k (t), & l \in \Omega, \end{cases}
\]

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, \( c_l (t + 1) \), is obtained from the states of its \( m + 1 \) neighboring nodes in \( \Theta_l \); and (iii) there is no central location and only local information is available.

**DILOC: Matrix format.** For compactness of notation, we write DILOC (10) in matrix form. Without loss of generality, we index the anchors in \( \kappa \) as \( 1, 2, \ldots, m + 1 \) and the sensors in \( \Omega \) as \( m + 2, m + 3, \ldots, m + 1 + M = N \). We now stack the (row vectors) states, \( c_i \), given in (4) for all the \( N \) nodes in the network in the \( N \times m \)-dimensional coordinate matrix

\[
C = \begin{bmatrix} c_1^T, & \ldots, & c_N^T \end{bmatrix}^T.
\]

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

\[
C(t + 1) = \Upsilon C(t).
\]
The structure of the $N \times N$ iteration matrix $\Upsilon$ becomes more apparent if we partition it as

$$
\Upsilon = \begin{bmatrix} I_{m+1} & 0 \\ B & P \end{bmatrix},
$$

(13)

The first $m+1$ rows correspond to the update equations for the anchors in $\kappa$. Since the states of the anchors are constant, see (B1) and (8), the first $m+1$ rows of $\Upsilon$ are zero except for a 1 at their $(q, q)\text{, } q \in \kappa = \{1, \ldots, m+1\}$ location. In other words, these first $m+1$ rows are the $(m+1)\times N$ block matrix $[I_{m+1}|0]$, i.e., the $(m+1)\times (m+1)$ identity matrix $I_{m+1}$ concatenated with the $(m+1)\times M$ zero matrix, $0$.

Each of the $M$ remaining rows in $\Upsilon$, indexed by $l \in \Omega = \{m+2, m+3, \ldots, N\}$, have only $m+1$ non-zero elements corresponding to the nodes in the triangulation set, $\Theta_l$, of $l$, and these non-zero elements are the barycentric coordinates, $a_{lk}$, of sensor $l$ with respect to the nodes in $\Theta_l$. The $M \times (m+1)$ block $B = \{b_{lj}\}$ is a zero matrix, except in those rows corresponding to the sensors in $\Omega$ that have a direct link to anchors. The $M \times M$ block $P = \{p_{lj}\}$ is also a sparse matrix where the non-zero entries in row $l$ correspond to the non-anchor nodes in $\Theta_l$. The matrices $\Upsilon$, $P$, and $B$ 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 $C(t)$ to $C(t+1)$. It is distributed because each row equation updates the state of sensor $l$ from the states of the $m+1$ nodes in $\Theta_l$. In all, the iteration matrix, $\Upsilon$, is highly sparse having exactly $(m+1) + M(m+1)$ non-zeros out of possible $(m+1+M)^2$ elements.

**C. Example**

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 set, $\Theta_l \text{, } l \in \Omega$, identified by using the convex hull inclusion test are $\Theta_4 = \{1, 5, 7\}, \Theta_5 =$
\[ \{4, 6, 7\}, \; \Theta_6 = \{2, 5, 7\}, \; \Theta_7 = \{3, 4, 6\}. \]

It is clear that these triangulation sets satisfy properties in (3). It can be verified that sensor 5 does not have any anchor node in its triangulation set, \( \Theta_5 \), and every other sensor has only one anchor in its respective triangulation set. Since, no sensor is able to communicate to \( m + 1 = 3 \) 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 inter-sensor distances (among the nodes in the set \( \{l\} \cup \Theta_l \)) in the Cayley-Menger determinant. It is noteworthy that the inter-sensor distances that need to be known at each sensor \( l \) to compute \( a_{lk} \) are only the inter-sensor distances among the \( m + 2 \) sensors 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 inter-sensor distances among the nodes in the set, \( \{5\} \cup \Theta_5 \), i.e., \( d_{54}, d_{56}, d_{57}, d_{46}, d_{47}, d_{67} \). Due to (3) and the relation \( \Theta_5 \subseteq K(l, r_5) \), the nodes in \( \{5\} \cup \Theta_5 \) lie in a circle of radius, \( r_5 \), centered at sensor 5 (shown in Fig. 1); on the other hand, no two sensors in the set \( \{5\} \cup \Theta_5 \) can be more than \( R_5 = 2r_5 \) apart. This justifies the choices of \( R_l \) and \( r_l = R_l/2 \) in (2).

Once the barycentric coordinates, \( a_{lk} \), are computed, DILOC for the sensors in \( \Omega \) is

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

In particular, for sensor 5, we have the following expression,

\[
\mathbf{c}_5(t + 1) = a_{54} \mathbf{c}_4(t) + a_{56} \mathbf{c}_6(t) + a_{57} \mathbf{c}_7(t).
\]

DILOC for the anchors is given by

\[
\mathbf{c}_q(t) = \mathbf{c}_q^*, \quad q \in \kappa.
\]

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

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

where the initial condition are \( \mathbf{C}(0) = [\mathbf{c}_1^0, \mathbf{c}_2^*, \mathbf{c}_3^*, \mathbf{c}_4^0, \mathbf{c}_5^0, \mathbf{c}_6^0, \mathbf{c}_7^0]^T \), 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.
D. 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 set-up phase, which identifies $\Theta_l(r_l)$ and computes the barycentric coordinates, as explained in Subsection 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 explicitly for it.

E. Random Poisson Deployment

A common model in wireless sensor networks is the Poisson deployment [27], [28]. We illustrate it on the plane, $m = 2$; the discussion can be extended to arbitrary dimensions. For a Poisson distribution with density, $\gamma > 0$, the mean number of sensors in a sector $Q$ with area $A_Q$ is $\gamma A_Q$. The numbers of sensors in any two disjoint sectors, $Q_1$ and $Q_2$, are independent random variables, and the locations of the sensors in a sector $Q$ are uniformly distributed. We now characterize the probability of triangulating a sensor $l$ in a region of radius, $r_l$, centered at $l$. To this end, consider Fig. 2(a), which shows the triangulation region, a circle of radius, $r_l$, centered at sensor $l$. Let

![Diagram](image)

Fig. 2. (a) Sensor $l$ identifies its triangulation set, $\Theta_l$, 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 sensor in each of these four sectors. (b) Illustration of Lemma 2.

$Q_1, Q_2, Q_3, Q_4$ be four disjoint sectors partitioning this circle with equal areas, i.e., $A_{Q_i} = \frac{\pi r_l^2}{4}$, $i = 1, \ldots, 4$.

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

3By sensor, we usually mean a non-anchor node.
Proof: In Fig. 2(b) consider the triangulation of sensor $l$ located at the center of the circle; we choose arbitrarily four sensors $p_1, p_2, p_3, p_4$ 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 $\text{Pol}(p_1p_2p_3p_4)$. Consider the diagonal $p_1p_3$ that partitions 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 $\triangle p_1p_2p_3 \cap \triangle p_1p_3p_4 = \emptyset$, then either $l \in \triangle p_1p_2p_3$ or $l \in \triangle p_1p_3p_4$. The triangle in which $l$ lies becomes the triangulating set, $\Theta_l$, 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 sensor in each of these eight sectors (with equal volume) of a sphere of radius, $r_l$, centered around $l$. ■

Let $Q_i$, be the set of sensors in the sector $Q_i$. It follows from the Poisson deployment that the probability of finding at least one sensor in a sector, $Q_i$, of area $A_{Q_i}$, is

$$P\left(|Q_i| > 0\right) = \left(1 - \exp^{-\gamma A_{Q_i}}\right). \tag{16}$$

Since the distribution of the number of sensors in disjoint sectors is independent, the probability of finding at least one sensor in each of the sets, $Q_1, \ldots, Q_4$, is the product

$$P\left(|Q_i| > 0, \forall i\right) = \left(1 - \exp^{-\gamma r_i^2/4}\right)^4. \tag{17}$$

Thus, we have

$$P_{\Theta_i} = P\left(\Theta_i \text{ exists satisfying (3)}\right) \geq P\left(|Q_i| > 0, \forall i\right). \tag{18}$$

This shows that, for a given deployment density, $\gamma$, we can choose, $r_i$, appropriately, to guarantee the triangulation with arbitrarily high probability. Indeed, it follows from (18) that, for an arbitrary, $0 < \epsilon < 1$, to get the probability of triangulation to be greater than $\epsilon$, i.e., $P_{\Theta_i} \geq \epsilon$, the radius $r_i$ should be

$$R_i \geq 2 \left(\frac{-4 \ln \left(1 - \epsilon^\frac{1}{4}\right)}{\gamma \pi}\right)^\frac{1}{2}. \tag{19}$$

In alternative, if we are limited by the communication radius, $R_i$, to guarantee $P_{\Theta_i} \geq \epsilon$, we will need the deployment density, $\gamma$, to be larger than

$$\gamma \geq \frac{-4}{\pi (R_i/2)^2} \ln \left(1 - \epsilon^\frac{1}{4}\right). \tag{20}$$

For example, if the sensors are deployed (following a Poisson distribution) with a density of $\gamma = 1$ sensor/m$^2$, then we can compute from equations (19)–(20) that 99% of the sensors will be able to triangulate (identify $\Theta_i$) themselves if they can communicate to at least a radius of $R_i = 5.52$ m. The remaining (1%) of the sensors may require to communicate to a larger radius.

4If $\text{Pol}(p_1p_2p_3p_4)$ is concave, we choose the diagonal that lies inside $\text{Pol}(p_1p_2p_3p_4)$, i.e., $p_1p_3$ in Fig. 2(b). If $\text{Pol}(p_1p_2p_3p_4)$ is convex, we can choose any of the two diagonals and the proof follows.
It also follows from the above discussion that if the sensors are equipped with a sense of directionality (for example, if they all have ultrasound transducers) then each sensor has to find one neighbor in each of its 4 sectors, $Q_{l,1}, Q_{l,2}, Q_{l,3}, Q_{l,4}$ (in $m = 2-d$ space). Once a neighbor is found, triangulation chooses 3 out of these 4, in order to identify $\Theta_l$. The computational complexity in $m = 2-d$ Euclidean space is $\binom{4}{3} = 4$. Without directionality the process of finding $\Theta_l$ has the (expected) computation complexity of $\gamma \pi r_l^2$ choose 3.

III. CONVERGENCE OF DILOC

In this section, we prove the convergence of DILOC to the exact locations of the sensors, $c^*_l, l \in \Omega$. To formally state the convergence result, we provide briefly some background and additional results, based on assumptions (B0)–(B3).

The entries of the rows of the iteration matrix $\Upsilon$, in (12), are either zero or the barycentric coordinates, $a_{lk}$, which are positive, and, by (7), add to 1. This matrix can then be interpreted as the transition matrix of a Markov chain. We then describe 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 let the $(ij)$-th element of the iteration matrix, $\Upsilon = \{\upsilon_{ij}\}$ defines the probability that the $i$th state goes to the $j$th state.

Because of the structure of $\Upsilon$, this chain is a very special Markov chain.

Absorbing Markov chain. Let an $N \times N$ matrix, $\Upsilon = \{\upsilon_{ij}\}$, 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 non-zero probability to an absorbing state (not necessarily in one step). In an absorbing Markov chain, a state which is not absorbing is called transient. For additional background, see, for example, [29].

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

Proof: We prove by contradiction. Since $\upsilon_{ii} = 1, i \in \kappa$, the anchors are the absorbing states of the Markov chain. Since $\upsilon_{ii} = 0, i \in \Omega$, the (non-anchor) sensors are the transient states. Partition the transient states into two clusters C1 and C2, such that each transient state in C1 can go with non-zero probability to at least one of the absorbing states and, with probability 1, the transient states in C2 cannot reach an absorbing state. It follows that with probability 1 the transient states in C2 cannot reach the transient states in C1 (in one or multiple steps); otherwise, they reach an absorbing state with a non-zero probability. Let’s consider the lie on the boundary of the convex hull, $C(C2)$, i.e., the vertices of $C(C2)$. Because they are on boundary, they cannot lie in the interior of the convex hull any subset of sensors in $C(C2)$, and, thus, cannot triangulate themselves, which contradicts Lemma 1 and assumption (B3). In order to triangulate the boundary sensors in $C(C2)$, the boundary sensors in C2 must be able to reach the transient states and/or the absorbing states, that is to say that the boundary sensors in $C(C2)$ have to reach the sensors in C1 to be able to triangulate themselves. Hence, the Markov chain is absorbing.

August 7, 2008 DRAFT
Consider the partitioning of the iteration matrix, $\Upsilon$, in (13). With the Markov chain interpretation, the $M \times (m+1)$ block $B = \{b_{lj}\}$ is a transition probability matrix for the transient states to reach the absorbing states in one-step, and the block $M \times M$ $P = \{p_{lj}\}$ is a transition probability matrix for the transient states. With (13), $\Upsilon^{t+1}$ can be written as

$$
\Upsilon^{t+1} = \begin{bmatrix}
I_{m+1} & 0 \\
\sum_{k=0}^{t} P^k B & P^{t+1}
\end{bmatrix},
$$

(21)

and, as $t$ goes to infinity, we have

$$
\lim_{t \to \infty} \Upsilon^{t+1} = \begin{bmatrix}
I_{m+1} & 0 \\
(I_M - P)^{-1} B & 0
\end{bmatrix},
$$

(22)

by Lemmas 5 and 6, in appendix III. Lemmas 5 and 6 use the fact that if $P$ is the matrix associated to the transient states of an absorbing Markov chain, then $\rho(P) < 1$, where $\rho(\cdot)$ is the spectral radius of a matrix. With (22), DILOC (10) converges to

$$
\lim_{t \to \infty} C(t + 1) = \begin{bmatrix}
I_{m+1} & 0 \\
(I_M - P)^{-1} B & 0
\end{bmatrix} C(0).
$$

(23)

In (23), the coordinates of the $M$ sensors in $\Omega$ (last $M$ rows of $C(t + 1)$) are written as a function of the $m + 1$ anchors in $\kappa$ whose coordinates are exactly known. The limiting values of the states of the $M$ sensors in $\Omega$ are written in terms of the coordinates of the $m + 1$ anchors in $\kappa$ weighted by $(I_M - P)^{-1} B$. To show that the limiting values are indeed the exact solution, we give the following Lemma.

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

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

(24)

Then, we have

$$
D = (I_M - P)^{-1} B.
$$

(25)

**Proof:** Clearly $(I_M - P)$ is invertible, since, by (95) in Appendix III, $\rho(P) < 1$; this follows from the fact that the eigenvalues of the matrix $I_M - P$ are $1 - \lambda_j$, where $\lambda_j$ is the $j$th eigenvalue of the matrix $P$ and $|\lambda_j| < 1, \forall j = 1, \ldots, M$. It suffices to show that,

$$
D = B + PD,
$$

(26)

since (25) follows from (26). In (26), $D$ and $B$ are both $M \times (m + 1)$ matrices, whereas $P$ is an $M \times M$ matrix whose non-zero elements are the barycentric coordinates for the sensors in $\Omega$. Hence, for the $lj$-th element in (26),
we need to show that
\[ d_{lj} = b_{lj} + \sum_{k \in \Omega} p_{lk} d_{kj}. \] (27)

For an arbitrary sensor, \( l \in \Omega \), its triangulation set, \( \Theta_l \), may contain nodes from both \( \kappa \) and \( \Omega \). We denote \( \kappa_{\Theta_l} \) as the elements of \( \Theta_l \) that are anchors, and \( \Omega_{\Theta_l} \) as the elements of \( \Theta_l \) that are non-anchor sensors. The exact coordinates, \( c^*_l \), 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.,
\[
c^*_l = \sum_{k \in \Theta_l} a_{lk} c^*_k,
\]
\[
= \sum_{j \in \kappa_{\Theta_l}} a_{lj} c^*_j + \sum_{k \in \Omega_{\Theta_l}} a_{lk} c^*_k,
\]
\[
= \sum_{j \in \kappa} b_{lj} c^*_j + \sum_{k \in \Omega} p_{lk} c^*_k, \quad (28)
\]
since the scalars, \( a_{lj} \), are given by
\[
a_{lj} = \begin{cases} 
  b_{lj}, & \text{if } j \in \kappa_{\Theta_l}, \\
  p_{lj}, & \text{if } j \in \Omega_{\Theta_l}, \\
  0, & \text{if } j \notin \Theta_l.
\end{cases} \quad (29)
\]

Equation (28) becomes, after writing each \( k \in \Omega \) in terms of the \( m + 1 \) anchors in \( \kappa \),
\[
c^*_l = \sum_{j \in \kappa} b_{lj} c^*_j + \sum_{k \in \Omega} p_{lk} \sum_{j \in \kappa} d_{kj} c^*_j,
\]
\[
= \sum_{j \in \kappa} b_{lj} c^*_j + \sum_{j \in \kappa} \sum_{k \in \Omega} p_{lk} d_{kj} c^*_j,
\]
\[
= \sum_{j \in \kappa} \left( b_{lj} + \sum_{k \in \Omega} p_{lk} d_{kj} \right) c^*_j. \quad (30)
\]

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 non-negative with their sum over \( j \in \kappa \) being 1 and the fact that the barycentric representation is unique, we must have
\[
d_{lj} = b_{lj} + \sum_{k \in \Omega} p_{lk} d_{kj}, \quad \text{which, comparing to (24), completes the proof.} \]

We now recapitulate these results in the following theorem.

**Theorem 1 (DILOC convergence)** DILOC (10) converges to the exact coordinates, \( c^*_l \), of the \( M \) sensors (with unknown locations) in \( \Omega \), i.e.,
\[
\lim_{t \to \infty} c_l(t+1) = c^*_l, \quad \forall \ l \in \Omega. \quad (32)
\]
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 \( P \), which by (95) in Appendix III is strictly less than one. This is a consequence of the fact that \( P \) is a sub-stochastic matrix. The convergence is slow if the spectral radius, \( \rho(P) \), is close to 1. This can happen if the matrix \( B \) is close to a zero matrix, \( 0 \). This is the case if and only 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 \( \kappa_{\Theta_l} \) are close to zero. Since in practical wireless sensor applications the sensors are assumed to be deployed in a geometric or a Poisson fashion (see details in Section II-E), the probability of this to happen is arbitrarily close to 0.

IV. DILOC WITH RELAXATION

In this Section, we modify DILOC to speed its convergence rate and 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 \( c_l(t + 1), \ l \in \Omega \), does not involve its own coordinates, \( c_l(t) \), at time \( t \). We introduce a relaxation parameter, \( \alpha \in (0, 1] \), in the iterations, such that, the expression of \( c_l(t + 1) \) is a convex combination of \( c_l(t) \) and (10). We refer to this version as the DILOC with relaxation, DILOC-REL. It is given by

\[
c_l(t + 1) = \begin{cases} 
(1 - \alpha)c_l(t) + \alpha c_l(t) = c_l(t), & \ l \in \kappa, \\
(1 - \alpha)c_l(t) + \alpha \sum_{k \in \Theta_l} a_{lk} c_k(t), & \ l \in \Omega.
\end{cases}
\]

(33)

DILOC is the special case of DILOC-REL with \( \alpha = 1 \). The matrix representation of DILOC-REL is

\[
C(t + 1) = HC(t),
\]

(34)

where \( H = (1 - \alpha)I_N + \alpha Y \) and \( I_N \) is the \( N \times N \) identity matrix. It is straightforward to show that the iteration matrix, \( H \), 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 \( J = (1 - \alpha)I_M + \alpha P \); partitioning \( H \) as

\[
H = \begin{bmatrix} I_{m+1} & 0 \\ \alpha B & J \end{bmatrix},
\]

(35)

We note the following

\[
H^{t+1} = \begin{bmatrix} I_{m+1} & 0 \\ \sum_{k=0}^{t} J^k \alpha B & J^{t+1} \end{bmatrix},
\]

(36)

and, as \( t \to \infty \),

\[
\lim_{t \to \infty} H^{t+1} = \begin{bmatrix} I_{m+1} & 0 \\ (I_M - J)^{-1} \alpha B & 0 \end{bmatrix},
\]

(37)
from Lemmas 5 and 6. Lemmas 5 and 6 apply to $H$, since $H$ is non-negative and $\rho(J) < 1$. To show $\rho(J) < 1$, we recall that $\rho(P) < 1$ and the eigenvalues of $J$ are $(1 - \alpha) + \alpha \lambda_j$, where $\lambda_j$ are the eigenvalues of $P$. Therefore, we have

$$\rho(J) = \max_j |(1 - \alpha) + \alpha \lambda_j| < 1.$$  \hspace{1cm} (38)

The following Theorem establishes convergence of DILOC-REL.

**Theorem 2** DILOC-REL (33) converges to the exact coordinates, $c_i^*$, of the $M$ sensors (with unknown locations) in $\Omega$, i.e.,

$$\lim_{t \to \infty} c_l(t + 1) = c_i^*, \quad \forall \ l \in \Omega.$$  \hspace{1cm} (39)

**Proof:** It suffices to show that

$$(I_M - J)^{-1} \alpha B = (I_M - P)^{-1} B.$$  \hspace{1cm} (40)

To this end, we note that

$$(I_M - J)^{-1} \alpha B = (I_M - ((1 - \alpha)I_M + \alpha P))^{-1} \alpha B,$$  \hspace{1cm} (41)

which reduces to (40) after basic algebraic manipulations. The convergence of DILOC-REL thus follows from Lemma 4.

As mentioned, the advantage of DILOC-REL is twofold: since $\rho(J)$ is a function of $\alpha$, we may optimize the convergence rate over $\alpha$; and DILOC-REL forms 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 inter-sensor distances are known up to errors, the communication links between sensors may fail and, when alive, the communication among sensors is corrupted by noise. We write the update equations for DILOC-REL, (34), in terms of the columns, $c_j(t)$, $1 \leq j \leq m$, of the coordinate matrix, $C(t)$. Column $j$ corresponds to the vector of the $j$-th estimate coordinates of all the $N$ sensor locations$^5$. The updates are

$$c_j(t + 1) = [(1 - \alpha)I + \alpha Y] c_j(t), \quad 1 \leq j \leq m.$$  \hspace{1cm} (42)

We partition $c_j(t)$ as

$$c_j(t) = \begin{bmatrix} u^j \\ x^j(t) \end{bmatrix},$$  \hspace{1cm} (43)

$^5$In the sequel, we omit the subscripts from the identity matrix, $I$, and its dimensions will be clear from the context.
where, \( u^j \in \mathbb{R}^{(m+1) \times 1} \) corresponds to the \( j \)-th coordinates of the anchors, which are known (hence, we omit the time index, as they are not updated), and \( x^j(t) \in \mathbb{R}^{M \times 1} \) corresponds to the estimates of the \( j \)-th coordinates of the non-anchor sensors, hence not known. Since, update is performed only on the \( x^j(t) \), (42) is equivalent to the following recursion:

\[
x^j(t + 1) = [(1 - \alpha)I + \alpha P]x^j(t) + \alpha Bu^j.
\] (44)

Thus, to implement the sequence of iterations in (44) perfectly, the \( l \)-th sensor at iteration \( t \) needs the corresponding rows of the matrices \( P \) and \( B \), and, in addition, the current estimates, \( c^j_n(t), n \in \Theta_l \) (\( j \)-th component of the \( n \)-th sensor coordinates), of its neighbors’ positions. In practice, there are several limitations: (i) The computation of the matrices \( P \) and \( B \) requires inter-sensor 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 (44) 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 \( B \) and \( P \), which in turn, depend on the inter-sensor distance measurements, which can be, possibly, random. Since a single measurement of the inter-sensor 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 \( l \)-th sensor can only get estimates, \( \hat{B}_l(t) \) and \( \hat{P}_l(t) \), of the corresponding rows of the \( B \) and \( P \) matrices, respectively. In the generic imperfect communication case, we have

\[
\hat{B}(t) = B + S_B + \tilde{S}_B(t)
\] (45)

where \( \{\tilde{S}_B(t)\}_{t \geq 0} \) is an independent sequence of random matrices with,

\[
\mathbb{E}\left[\tilde{S}_B(t)\right] = 0, \, \forall t, \quad \sup_{t \geq 0} \mathbb{E}\left[\|\tilde{S}_B(t)\|^2\right] = k_B < \infty.
\] (46)

Here, \( S_B \) is the mean measurement error. Similarly, for \( P \), we have

\[
\hat{P}(t) = P + S_P + \tilde{S}_P(t),
\] (47)

where \( \{\tilde{S}_P(t)\}_{t \geq 0} \) is an independent sequence of random matrices with,

\[
\mathbb{E}\left[\tilde{S}_P(t)\right] = 0, \, \forall t, \quad \sup_{t \geq 0} \mathbb{E}\left[\|\tilde{S}_P(t)\|^2\right] = k_P < \infty.
\] (48)

Likewise, \( S_P \) is the mean measurement error. Note that this way of writing \( \hat{B}(t), \hat{P}(t) \) 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 eqns. (46,48) are very weak and, in
In contrast with DILOC-REL, in (51), the gain environment case.

(C2) Random Link Failure: We assume that the inter-sensor 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 $\left\{ v_{ln}^j(t) \right\}_{l,n,j,t}$ be a family of independent zero mean random variables such that

$$\sup_{l,n,j,t} \mathbb{E}[v_{ln}^j(t)]^2 = k_v < \infty. \quad (49)$$

We assume that, at the $t$-th 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, $c_{ln}^j(t)$, given by

$$y_{ln}^j(t) = c_{ln}^j(t) + v_{ln}^j(t). \quad (50)$$

This models the channel noise. The moment assumption in eqn. (49) is very weak and holds, in particular, if the channel noise is i.i.d.

(C4) Independence: We assume that the sequences, $\left\{ \tilde{S}_B(t), \tilde{S}_P(t) \right\}_{t \geq 0}$, $\left\{ e_{ln}(t) \right\}_{l,n,t}$, and $\left\{ v_{ln}^j(t) \right\}_{l,n,j,t}$ 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.

Clearly, under the random environment model (as detailed in Assumptions (C1)-(C4), the algorithm in (44) is not appropriate to update the sensors states. We now consider the following state update recursion for the random environment case.

**Distributed Localization in Random Environment Algorithm (DLRE):**

$$\begin{align*}
x_{ln}^j(t+1) &= (1 - \alpha(t)) x_{ln}^j(t) + \alpha(t) \left[ \sum_{n \in \Omega \cap \Theta_l} \frac{e_{ln}(t) \tilde{B}_{ln}(t)}{q_{ln}} \left( w_{ln}^j + v_{ln}^j(t) \right) \right] \\
&+ \alpha(t) \left[ \sum_{n \in \Omega \cap \Theta_l} \frac{e_{ln}(t) \tilde{P}_{ln}(t)}{q_{ln}} \left( x_{ln}^j(t) + v_{ln}^j(t) \right) \right], \; l \in \Omega, \; 1 \leq j \leq m
\end{align*} \quad (51)$$

In contrast with DILOC-REL, in (51), the gain $\alpha(t)$ is now time varying. It will become clear why when we study convergence of this algorithm. To write DLRE in a compact form, we introduce notation. Define the random matrices, $\tilde{B}(t) \in \mathbb{R}^{M \times (m+1)}$ and $\tilde{P}(t) \in \mathbb{R}^{M \times (m+1)}$, as the matrices with ln entries given by

$$\tilde{B}_{ln}(t) = \tilde{B}_{ln}(t) \left( \frac{e_{ln}(t)}{q_{ln}} - 1 \right), \quad \tilde{P}_{ln}(t) = \tilde{P}_{ln}(t) \left( \frac{e_{ln}(t)}{q_{ln}} - 1 \right). \quad (52)$$
Clearly, by (C2)-(C4), the matrices $\tilde{B}(t) \in \mathbb{R}^{M \times (m+1)}$ and $\tilde{P}(t) \in \mathbb{R}^{M \times (m+1)}$ are zero mean. Note that $\mathbb{E}[e_{ln}] = q_{ln}$. Also, by the bounded moment assumptions in (C1), we have

$$\sup_{t \geq 0} \mathbb{E} \left[ \left\| \tilde{B}(t) \right\|^2 \right] = k_B < \infty, \quad \sup_{t \geq 0} \mathbb{E} \left[ \left\| \tilde{P}(t) \right\|^2 \right] = k_P < \infty. \quad (53)$$

Hence, the iterations in (51) can be written in vector form as

$$x^j(t+1) = (1 - \alpha(t)) x^j(t) + \alpha(t) \left[ \left( \tilde{P}(t) + \tilde{P}(t) \right) x^j(t) + \left( \tilde{B}(t) + \tilde{B}(t) \right) u^j + \eta^j(t) \right], \quad (54)$$

where, the $l$th element of the vector, $\eta^j(t)$, is given by

$$\eta^j(t) = \sum_{n \neq l} \left( \tilde{P}_{ln}(t) + \tilde{P}_{ln}(t) \right) v^j_{ln}(t) + \sum_{n \neq l} \left( \tilde{B}_{ln}(t) + \tilde{B}_{ln}(t) \right) v^j_{ln}(t). \quad (55)$$

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

$$\sup_{t} \mathbb{E} \left[ \left\| \eta^j(t) \right\|^2 \right] = k_{\eta} < \infty. \quad (56)$$

From (C1), the iteration sequence in (54) can be written as

$$x^j(t+1) = x^j(t) - \alpha(t) \left[ (I - P - S_P) x^j(t) - (B + S_B) u^j - (S_P(t) + \tilde{P}(t)) x^j(t) \right. \right.$$

$$\left. - \left( S_B(t) + \tilde{B}(t) \right) u^j - \eta^j(t) \right]. \quad (57)$$

We now make two additional design assumptions.

**(D1) Persistence Condition:** The weight sequence satisfies

$$\alpha(t) > 0, \quad \sum_{t \geq 0} \alpha(t) = \infty, \quad \sum_{t \geq 0} \alpha^2(t) < \infty. \quad (58)$$

This condition, commonly assumed in the adaptive control and adaptive signal processing literature, assumes that the weights decay to zero, but not too fast.

**(D2) Low Error Bias:** We assume that

$$\rho (P + S_P) < 1. \quad (59)$$

Clearly, we have $\rho(P) < 1$. Thus, if we assume that the non-zero bias, $S_P$, in the system matrix (resulting from incorrect distant computation) is small, (59) is justified. We note that this condition ensures that the matrix $(I - P - S_P)$ is invertible.

In the following sections, we prove that the DLRE algorithm, under the assumptions (C1)-(C4), (D1)-(D2), leads to a.s. convergence of the state vector sequence, $\{x^j(t)\}_{t \geq 0}$, to a deterministic vector for each $j$, which may be different from the exact sensor locations, because of the random errors in the iterations. We characterize this error and show that it depends on the non-zero biases, $S_B$, and $S_P$ in the system matrix computations, and vanishes as $\|S_B\| \to 0$ and $\|S_P\| \to 0$. 

August 7, 2008
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 \( \{x^j(t)\}_{t \geq 0}, 1 \leq j \leq m \), be the state sequence generated by the iterations, given by (57), under the assumptions (C1)-(C4), (D1)-(D2). Then,

\[
P \left[ \lim_{t \to \infty} x^j(t) = (I - P - SP)^{-1}(B + SP)u^j, \forall j \right] = 1.
\] (60)

The convergence analysis of the DLRE algorithm 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., [30], [31]. The proof relies on the following result from [32], which we state here as a theorem.

Theorem 4 Consider the following recursive procedure:

\[
x(t+1) = x(t) + \alpha(t) \left[R(x(t)) + \Gamma(t+1, x(t), \omega)\right],
\] (61)

where, \( x, R, \Gamma \) are vectors in \( \mathbb{R}^{M \times 1} \). There is an underlying common probability space \( (\Xi, F, P) \), and \( \omega \) is the canonical element of the probability space, \( \Xi \). Assume that the following conditions are satisfied:

1): The vector function \( R(x) \) is Borel measurable and \( \Gamma(t, x, \omega) \) is \( B^M \otimes F \) measurable for every \( t \).

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

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

\[
V(x_0) = 0, \quad V(x) > 0, \quad x \neq x_0, \quad \sup_{\|x-x_0\| > \epsilon} \langle R(x), V_x(x) \rangle < 0, \quad \forall \epsilon > 0.
\] (63)

4): There exist constants \( k_1, k_2 > 0 \), such that,

\[
\|R(x)\|^2 + E[\|\Gamma(t, x, \omega)\|^2] \leq k_1(1 + V(x)) - k_2 \langle R(x), V_x(x) \rangle
\] (64)

5): The weight sequence \( \{\alpha(t)\}_{t \geq 0} \) satisfies the persistence condition (D1) given by (58).

Then the Markov process, \( \{x(t)\}_{t \geq 0} \), converges a.s. to \( x_0 \).

Proof: The proof follows from Theorem 4.4.4 in [32] and is omitted due to space constraints.

We now return to the proof of Theorem 3.

\(^6\text{In the sequel, } B^M \text{ denotes the Borel sigma algebra in } \mathbb{R}^{M \times 1}. \text{ The space of twice continuously differentiable functions is denoted by } \mathbb{C}_2, \text{ while } V_x(x) \text{ denotes the gradient } \frac{\partial V(x)}{\partial x}.\)
Proof: [Proof of Theorem 3] We will show that, under the assumptions, the algorithm in (57) falls under the purview of Theorem 4. To this end, consider the filtration, \( \{ \mathcal{F}_t \}_{t \geq 0} \), where

\[
\mathcal{F}_t = \sigma \left( x^j(0), \tilde{S}_P(s), \tilde{P}(s), \tilde{S}_B(s), \tilde{B}(s), \eta^j(s) : 0 \leq s < t \right).
\]

(65)

Define the vector \( d^* \) as

\[
d^* = (I - P - S_P)^{-1}(B + S_B)u^j.
\]

(66)

Equation (57) can be written as

\[
x^j(t + 1) = x^j(t) - \alpha(t) \left[ (I - P - S_P)(x^j(t) - d^*) - (\tilde{S}_P(t) + \tilde{P}(t))x^j(t) \right.
\]

\[
- \left. (\tilde{S}_B(t) + \tilde{B}(t))u^j - \eta^j(t) \right].
\]

(67)

In the notation of Theorem 4, (67) is given by

\[
x^j(t + 1) = x^j(t) + \alpha(t) \left[ R(x^j(t)) + \Gamma(t + 1, x^j(t), \omega) \right],
\]

(68)

where

\[
R(x) = -(I - P - S_P)(x^j(t) - d^*),
\]

(69)

and

\[
\Gamma(t + 1, x, \omega) = \left[ (\tilde{S}_P(t) + \tilde{P}(t))x^j(t) + (\tilde{S}_B(t) + \tilde{B}(t))u^j + \eta^j(t) \right].
\]

(70)

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. To this end, define

\[
V(x) = \|x - d^*\|^2.
\]

(71)

Clearly, \( V(x) \in C_2 \) with bounded second order partial derivatives, with

\[
V(d^*) = 0, \quad V(x) > 0, \quad x \neq d^*.
\]

(72)
Also, we note that, for \( \varepsilon > 0 \),

\[
\sup_{\|x - d_*\| > \varepsilon} \langle R(x), V(x) \rangle = \sup_{\|x - d_*\| > \varepsilon} -2(x - d^*)^T (I - P - S_p) (x - d^*)\,.
\]

\[
= \sup_{\|x - d_*\| > \varepsilon} \left[ 2(x - d^*)^T (P + S_p) (x - d^*) - 2\|x - d^*\|^2 \right],
\]

\[
\leq \sup_{\|x - d_*\| > \varepsilon} \left[ 2\|x - d^*\| \rho (P + S_p) \|x - d^*\| - 2\|x - d^*\|^2 \right],
\]

\[
= \sup_{\|x - d_*\| > \varepsilon} -2(1 - \rho (P + S_p)) \|x - d^*\|^2,
\]

\[
< -2\varepsilon^2 (1 - \rho (P + S_p)),
\]

\[
\leq 0,
\]

(73)

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

To verify 4) note that

\[
\|R(x)\|^2 = (x - d^*)^T (I - P - S_p)^T (I - P - S_p) (x - d^*),
\]

\[
\leq \| (I - P - S_p)^T (I - P - S_p) \| \|x - d^*\|^2,
\]

\[
= k_1 \|x - d^*\|^2,
\]

\[
= k_1 V(x),
\]

(74)

where \( k_1 > 0 \) is a constant.

Finally, by assumptions (C1)-(C4), we have

\[
\mathbb{E} \| \Pi(t, x, \omega) \|^2 = \mathbb{E} \left[ \left( \bar{S}_p(t - 1) + \bar{P}(t - 1) \right) x + \left( \bar{S}_b(t - 1) + \bar{B}(t - 1) \right) u' + \eta'(t - 1) \right]^T
\]

\[
\left[ \left( \bar{S}_p(t - 1) + \bar{P}(t - 1) \right) x + \left( \bar{S}_b(t - 1) + \bar{B}(t - 1) \right) u' + \eta'(t - 1) \right],
\]

\[
= x^T \mathbb{E} \left[ \bar{S}_p^T(t - 1) \bar{S}_p(t - 1) + \bar{P}^T(t - 1) \bar{P}(t - 1) \right] x + u'^T \mathbb{E} \left[ \bar{S}_b^T(t - 1) \bar{S}_b(t - 1) \right] u',
\]

\[
+ \bar{B}^T(t - 1) \bar{B}(t - 1) u'^T + \mathbb{E} \left[ \| \eta' (t - 1) \|^2 \right]^2 + 2x^T \mathbb{E} \left[ \bar{S}_b^T(t - 1) \bar{S}_b(t - 1) \right] u'^2,
\]

\[
\leq \mathbb{E} \left[ \| \bar{S}_p(t - 1) \|^2 + \| \bar{P}(t - 1) \|^2 \right] \|x\|^2 + \mathbb{E} \left[ \| \bar{S}_b(t - 1) \|^2 + \| \bar{B}(t - 1) \|^2 \right] \|u'\|^2
\]

\[
+ \mathbb{E} \left[ \| \eta' (t - 1) \|^2 \right] + 2 \mathbb{E} \left[ \| \bar{S}_p(t - 1) \|^2 \right] \|x\|^2 + \mathbb{E} \left[ \| \bar{S}_b(t - 1) \|^2 \right] \|u'\|^2 + 2x^T \mathbb{E} \left[ \bar{S}_b^T(t - 1) \bar{S}_b(t - 1) \right] u'^2,
\]

\[
\leq \left( k_P + k_B \right) \|x\|^2 + \left( k_B + k_B \right) \|u'\|^2 + k_\eta + k_p k_B \|x\| \|u'\|.
\]

(75)

The cross terms dropped in the second step of eqn. (75) have zero mean by the independence assumption (C4).

For example, consider the term \( \mathbb{E} \left[ \bar{S}_b^T(t - 1) \bar{P}(t - 1) \right] \). It follows from eqns. (47,52) that the \( h^n \)-th entry of the
matrix $\tilde{S}_P^T(t-1)\tilde{P}(t-1)$ is given by

$$\begin{align*}
[\tilde{S}_P^T(t-1)\tilde{P}(t-1)]_{rl} &= \sum_r [\tilde{S}_P^T(t-1)]_{rl} [\tilde{P}(t-1)]_{rn} \\
&= \sum_r [\tilde{S}_P(t-1)]_{rl} [\tilde{P}(t-1)]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \\
&= \sum_r [\tilde{S}_P(t-1)]_{rl} [\tilde{P}(t-1)]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) + \sum_r [\tilde{S}_P(t-1)]_{rl} [S_P]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right)
\end{align*}$$

(76)

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

$$\begin{align*}
E \left[ [S_P(t-1)]_{rl} [P]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] &= [P]_{rn} E \left[ [S_P(t-1)]_{rl} \right] E \left[ \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] = 0 \\
E \left[ [\tilde{S}_P(t-1)]_{rl} [S_P]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] &= [S_P]_{rn} E \left[ [\tilde{S}_P(t-1)]_{rl} \right] E \left[ \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] = 0 \\
E \left[ [\tilde{S}_P(t-1)]_{rl} [\tilde{S}_P(t-1)]_{rn} \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] &= E \left[ [S_P(t-1)]_{rl} [S_P(t-1)]_{rn} \right] E \left[ \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] = 0
\end{align*}$$

where we have repeatedly used the fact that

$$E \left[ \left( \frac{\varepsilon_{rn}(t)}{q_{rn}} - 1 \right) \right] = 0$$

(77)

From eqns. (76-77) it is then clear that

$$E \left[ \tilde{S}_P^T(t-1)\tilde{P}(t-1) \right] = 0$$

(78)

In a similar way, it can be shown that the other dropped crossed terms in eqn. (75) are zero-mean.

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

$$\|x\|^2 \leq k_3\|x - d^*\|^2 + k_4, \quad \|x\| \leq k_5\|x - d^*\|^2 + k_6.$$  

(79)

Hence, from (75) and (79), we have

$$\begin{align*}
E \|G(t, x, \omega)\|^2 &\leq k_7 \|x - d^*\|^2 + k_8, \\
&\leq k_9 (1 + V(x)),
\end{align*}$$

(80)

where, $k_7, k_8 > 0$ and $k_9 = \max (k_7, k_8)$. Combining eqns. (74,80) we note that assumption 4) in Theorem 4 is satisfied, as

$$\langle R(x), V_x(x) \rangle \leq 0, \ \forall x.$$  

(81)
Hence, all the conditions of Theorem 4 are satisfied and we conclude that

$$\mathbb{P} \left[ \lim_{t \to \infty} x^j(t) = (I - P - SP)^{-1} (B + SB) u^j \right] = 1. \quad (82)$$

Since, (82) holds for all $j$, and $j$ takes only a finite number of values ($1 \leq j \leq m$), we have

$$\mathbb{P} \left[ \lim_{t \to \infty} x^j(t) = (I - P - SP)^{-1} (B + SB) u^j, \ \forall j \right] = 1. \quad (83)$$

We now interpret Theorem 3. Referring to the partitioning of the $C(t)$ matrix in (43), we have

$$C(t) = \begin{bmatrix} U \\ X(t) \end{bmatrix}, \quad (84)$$

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

$$\mathbb{P} \left[ \lim_{t \to \infty} X(t) = (I - P - SP)^{-1} (B + SB) U \right] = 1. \quad (85)$$

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

$$X^* = (I - P)^{-1} BU. \quad (86)$$

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

$$e_l = \left\| (I - P - SP)^{-1} (B + SB) U - (I - P)^{-1} BU \right\|. \quad (87)$$

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

**VII. Numerical Studies**

We divide the numerical study of DILOC into the following parts. First, we present DILOC in the deterministic case, i.e., we have no communication noise, no link failures, and the required inter-sensor distances are known precisely. Second, we present DILOC when there is communication noise and link failures. Third, we consider noise on the distance measurements that results in random system matrices, $\tilde{P}(t)$ and $\tilde{B}(t)$, as given in (45)–(47);
we study both the biased $(S_P \neq 0, S_B \neq 0)$ and unbiased $(S_P = S_B = 0)$ cases in the following. In the end, we present studies of DILOC in the presence of all random scenarios.

**DILOC Algorithm in Deterministic Environments:** We consider the example presented in section II-C. We have $N = 7$ nodes in $m = 2$-dimensional space, where $m + 1 = 3$ are the anchors and $M = 4$ are the sensors. DILOC, as given in (14), is implemented, and the results are shown in Fig. 3(a)–3(b). 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 condition. Fig. 3(c) and Fig. 3(d) show DILOC for a network of $N = 500$ nodes.

**DILOC Algorithm with Communication Noise and Link Failures:** We consider the same example of section II-C, but, include noise in the communication and link failures. All the communication links are active 90% of the time, i.e., $q_{ln} = 0.9$, $\forall n$ s.t. $n \in \Theta_l$, as discussed in (C2), and include an additive communication noise that is Gaussian i.i.d. with zero-mean and variance $1/M$ (roughly speaking, this is equivalent to having a unity variance for the entire network). In this scenario, we employ DILOC with a decreasing weight sequence, $\alpha(t) = \frac{1}{t+1}$ and the results are presented in Fig. 4(a) and Fig. 4(b).

**DILOC with Noisy Distance measurements:** We now consider noise on the distance measurements. We assume that we have a reasonable estimate of the required distances such that it translates into a small perturbation of the
system matrices, $\hat{B}(t)$ and $\hat{P}(t)$. The matrices $\tilde{S}_P, \tilde{S}_B$ are zero-mean Gaussian i.i.d. perturbations with variance 0.1 (small signal perturbation, note that the non-zero elements of both the matrices, $B$ and $P$ lie in the range of 0 and 1). For a network of $N = 50$ nodes in Fig. 5(a), we implement DLRE, with a decreasing weight sequence, $\alpha = \frac{1}{1.55}$, in Fig. 5(b). Finally, Fig. 6(a) shows a network of $N = 50$ nodes, where DLRE, with a decreasing weight sequence, $\alpha = \frac{1}{1.55}$, with all of the above random scenarios is implemented in Fig. 6(b).

VIII. CONCLUSIONS

The paper studies a distributed iterative sensor localization algorithm 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 (sensors 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 no noise affects the inter-sensor communication, the inter-sensor distances are known with no errors, and the communication links do not fail, we show that our distributed algorithms, DILOC and DILOC-REL, lead to convergence to the exact sensor locations. For the random environment scenario, where inter-sensor communication links may fail randomly, transmitted data is distorted by noise, and inter-sensor distance information is imprecise, we show that our modified algorithm, DLRE, leads to almost sure convergence of the iterative location estimates, and 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.

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, \( \kappa \), using only the mutual distance information among these \( m+2 \) nodes (\( \kappa \cup \{l\} \)). Let \( \kappa \) denote the set of \( m+1 \) nodes and let \( C(\kappa) \) denote the convex hull formed by the nodes in \( \kappa \). Clearly, if \( l \in C(\kappa) \), then the convex hull formed by the nodes in \( \kappa \) is the same as the convex hull formed by the nodes in \( \kappa \cup \{l\} \), i.e.,

\[
C(\kappa) = C(\kappa \cup \{l\}), \quad \text{if } l \in C(\kappa).
\]  

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

\[
A_\kappa = A_{\kappa \cup \{l\}}, \\
= \sum_{k \in \kappa} A_{\kappa \cup \{l\} \setminus \{k\}}, \quad \text{if } l \in C(\kappa).
\]  

Hence, the test becomes

\[
l \in C(\kappa), \quad \text{if } \sum_{k \in \kappa} A_{\kappa \cup \{l\} \setminus \{k\}} = A_\kappa, \quad (90)
\]

\[
l \notin C(\kappa), \quad \text{if } \sum_{k \in \kappa} A_{\kappa \cup \{l\} \setminus \{k\}} > A_\kappa, \quad (91)
\]

This is also shown in Figure 7. 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-MENDER DETERMINANT

Let $\kappa$ be a set of $m+1$ points (sensors) in $\mathbb{R}^m$, and $d_{lj}$ be the inter-sensor distance between $l$ and $j$. The generalized volume, $A_\kappa$, of the convex hull of the points in $\kappa$ can be computed by the Cayley-Menger determinant, see, e.g., [33]. The Cayley-Menger determinant is the determinant of an $m+2 \times m+2$ (symmetric) matrix that relates to the generalized volume, $A_\kappa$, of the convex hull, $C(\kappa)$, of the $m+1$ points in $\mathbb{R}^m$ through an integer sequence, $s_{m+1}$. The Cayley-Menger determinant is given by

$$
\begin{vmatrix}
0 & 1 & 1 & 1 & \ldots & 1 \\
1 & 0 & d_{12}^2 & d_{13}^2 & \ldots & d_{1,m+1}^2 \\
1 & d_{21}^2 & 0 & d_{23}^2 & \ldots & d_{2,m+1}^2 \\
1 & d_{31}^2 & d_{32}^2 & 0 & \ldots & d_{3,m+1}^2 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
1 & d_{m+1,1}^2 & d_{m+1,1}^2 & \ldots & d_{m+1,m-1}^2 & d_{m+1,m}^2 & 0
\end{vmatrix},
$$

(92)

where

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

(93)

and its first few terms are $-1, 2, -16, 288, -9216, 460800, \ldots$.

APPENDIX III
IMPORTANT RESULTS

**Lemma 5** If the matrix, $P$, corresponds to the transition probability matrix associated to the transient states of an absorbing Markov chain, then

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

(94)

**Proof:** For such a matrix, $P$, we have

$$\rho(P) < 1,$$

(95)

from Lemma 8.3.20 and Theorem 8.3.21 in [34], where $\rho(\cdot)$ denotes the spectral norm of a matrix and (94) follows from (95).

**Lemma 6** If the matrix, $P$, corresponds to the transition probability matrix associated to the transient states of an absorbing Markov chain, then

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

(96)

**Proof:**
The proof follows from Lemma 5 and Lemma 6.2.1 in [34].

REFERENCES

[1] Usman A. Khan and José M. F. Moura, “Distributing the Kalman filters for large-scale systems,” IEEE Transactions on Signal Processing, Aug. 2007, accepted for publication, DOI: 10.1109/TSP.2008.927480.
[2] G. Springer, Introduction to Riemann surfaces, Addison-Wesley, Reading, MA, 1957.
[3] J. G. Hocking and G. S. Young, Topology, Addison–Wesley, Reading, MA, 1961.
[4] R. L. Moses, D. Krishnamurthy, and R. Patterson, “A self-localization method for wireless sensor networks,” EURASIP Journal on Applied Signal Processing, , no. 4, pp. 348–358, Mar. 2003.
[5] N. Patwari, A. O. Hero III, M. Perkins, N. Correal, and R. J. ODea, “Relative location estimation in wireless sensor networks,” IEEE Trans. on Signal Processing, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
[6] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity,” in 4th ACM international symposium on mobile ad-hoc networking and computing, Annapolis, MD, Jun. 2003, pp. 201–212.
[7] Y. Shang and W. Ruml, “Improved MDS-based localization,” in IEEE Infocom, Hong Kong, Mar. 2004, pp. 2640–2651.
[8] F. Thomas and L. Ros, “Revisiting trilateration for robot localization,” IEEE Transactions on Robotics, 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,” 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 IEEE International Conference on Sig. Proc., Montreal, Canada, Mar. 2004, vol. 3, pp. 857–860.
[12] D. Niculescu and B. Nath, “Ad-hoc positioning system,” in IEEE Globecom, 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 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 Intl. Workshop on Sensor Networks and 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 2nd Intl. Workshop on Information Processing in Sensor Networks, Palo Alto, CA, Apr. 2003, pp. 333–348.
[16] J. J. Caffery, Wireless location in CDMA cellular radio systems, Kluwer Academic Publishers, Norwell, MA, 1999.
[17] J. A. Costa, N. Patwari, and III A. O. Hero, “Distributed weighted-multidimensional scaling for node localization in sensor networks,” ACM Transactions on Sensor Networks, vol. 2, no. 1, pp. 39–64, 2006.
[18] J. Albowicz, A. Chen, and L. Zhang, “Recursive position estimation in sensor networks,” in IEEE Int. Conf. on 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 IEEE International Conference on Sig. Proc., Salt Lake City, UA, May 2001, pp. 2037–2040.
[20] S. Čapkun, M. Hamdi, , and J. P. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” in 34th IEEE Hawaii Int. Conf. on System Sciences, Wailea Maui, HI, Jan. 2001.
[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 IEEE International Conference on Sig. Proc., Montreal, Canada, May 2004.
[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 IEEE Information Processing in Sensor Networks, Berkeley, CA, Apr. 2004, pp. 99–107.
[24] S. Thrun, “Probabilistic robotics,” Communications of the ACM, vol. 45, no. 3, pp. 52–57, Mar. 2002.
[25] J. C. Gower, “Euclidean distance geometry,” Mathematical Scientist, vol. 7, pp. 1–14, 1982.
[26] J. C. Gower, “Properieties of Euclidean and non-Euclidean distance matrices,” Linear Algebra and its Applications, vol. 67, pp. 81–97, Jun. 1985.
[27] P. Hall, *Introduction to the theory of coverage processes*, John Wiley and Sons Inc., Chichester, UK, 1988.

[28] Y. Sung, L. Tong, and A. Swami, “Asymptotic locally optimal detector for large-scale sensor networks under the poisson regime,” *IEEE Transactions on Signal Processing*, vol. 53, no. 6, pp. 2005–2017, Jun. 2005.

[29] C. M. Grinstead and J. L. Snell, *Introduction to probability*, American Mathematical Society, 1997.

[30] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks: Link failures and channel noise,” November 2007, Manuscript submitted to the IEEE Transactions on Signal Processing, http://arxiv.org/abs/0711.3915.

[31] S. Kar and J. M. F. Moura, “Distributed consensus algorithms in sensor networks: Quantized data,” December 2007, Manuscript submitted to the IEEE Transactions on Signal Processing, http://aps.arxiv.org/abs/0712.1609.

[32] M.B. Nevel’son and R.Z. Has’minskii, *Stochastic approximation and recursive estimation*, American Mathematical Society, Providence, Rhode Island, 1973.

[33] M. J. Sippl and H. A. Scheraga, “Cayley–Menger coordinates,” *Proceedings of the National Academy of Sciences of U.S.A.*, vol. 83, no. 8, pp. 2283–2287, Apr. 1986.

[34] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*, Academic Press, INC., New York, NY, 1970.