IEEE Copyright Statement:

Forty-Sixth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 23-26, 2008

Distributed Sensor Localization in Euclidean Spaces: Dynamic Environments

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

***Abstract*— In [1], we presented an algorithm to localize sensors in m-dimensional Euclidean space $\mathbb{R}^n$ with unknown locations assuming the following: 1) there are $(m+1)$ sensors that know their absolute coordinates–the anchors; 2) each sensor communicates with $m+1$ of its neighbors; and 3) the sensors lie in the convex hull of the anchors. The localization algorithm is a generalization of consensus–it is a weighted linear, iterative, and distributed algorithm. The weights are the barycentric coordinates of a sensor with respect to its neighbors, which are computed by the generalized volumes obtained from the intersensor distances in the Cayley-Menger determinants. This paper expands on this work to take advantage of when the number of anchors available possibly exceeds $m+1$, a sensor can communicate with all sensors within its radius of communication, and when the network communication topology may be dynamic as, for example, when the network neighborhood structure changes over time. The paper shows that the algorithm converges to the exact sensor locations in the absence of noise.**

I. Introduction

Localization is a fundamental problem in sensor networks. In applications where the sensors are deployed randomly, information about their locations is key to place the sensor measurements in a proper geographical context. For this purpose, adding a GPS receiver to each sensor has several disadvantages: they are prohibitively expensive; not robust to jamming for military applications; and the satellite reception is restricted in the indoor environments. In [1], we presented a distributed sensor localization algorithm in the $m-$dimensional Euclidean space, $\mathbb{R}^m$ $(m \geq 1)$, that requires a minimal number, $m+1$, of anchors who know their locations exactly[1]. We assume that the sensors lie in the convex hull of the anchors. The resulting distributed localization algorithm is iterative and we show that the iterations converge to the exact sensor locations. At each sensor, the iterations involve a convex combination of the location estimates at the neighboring sensors. The weights of the convex combinations are computed using the barycentric coordinates, [2], [3], and the Cayley-Menger determinants, [4]. We note that the inclusion of an arbitrary number of anchors and arbitrary neighborhood sizes for each sensor generally leads to faster convergence of the localization algorithm, as shown by our detailed simulations.

The setup we assume in this paper (underwater, on the surface, or in freespace) suits most practical applications, for example: (i) M UAVs in $\mathbb{R}^3$ lying in the convex hull of a minimal of $m+1=4$ beacons. The beacons are the anchors and the M UAVs are the sensors. Each UAV communicates to its neighboring $m+1$ UAVs and only knows its distance to these neighbors. (ii) In a battlefield, M soldiers with unknown locations are positioned in the convex hull of $m+1$ base stations. The base stations serve as anchors and each base station communicates only with a few soldiers. The soldiers communicate with $m+1-$neighboring soldiers with respect to whom they know their mutual distances. Other applications include localizing sensors in a sensor network, packages in a shipping facility, unmanned aircrafts, objects in a warehousing/manufacturing facility, and animals to study their behavior and their interactions. A comprehensive study of several applications is in [5]. In all these cases, it is reasonable to assume that all the objects lie in the convex hull of the $m+1$ anchors and that the communications are local, i.e., they can only communicate with a few neighboring objects. A distributed localization algorithm is essential for such scenarios with the sensors interacting with neighboring sensors to iteratively learn their own location. In this work, we further extend these results to dynamic network topologies and more than $m+1$ anchors and neighbors.

Work on distributed localization algorithms include [6], [7], [8], [9], [10], [11], [12]. In [1], we provide a brief review of localization algorithms; we refer the reader to [1] and the references there in.

The algorithm we present here is of the class of consensus algorithms, see [13]. Consensus is a distributed iterative algorithm, [14], which has received considerable attention recently. Many applications reduce to consensus like dynamic load balancing, [15], multiagent coordination, [16], distributed detection, [17], [18], for a recent review see [19]. In particular, in distributed detection, consensus extends the parallel architecture traditionally used, [20], [21], [22], for a review of early work on parallel fusion see [23], and for more recent work, see our own and the references therein, [24], [25], [26], to arbitrary network topologies.

We organize the rest of the paper as follows. Section II overviews the localization algorithm presented in [1]. Sec-

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.

The authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213. {ukhan, brunos, moura}@ece.cmu.edu, soummyak@andrew.cmu.edu

[1]It is well-known that resolving the location of an arbitrary sensor (in $\mathbb{R}^m$) using the sensor-anchor distances requires the sensor to communicate to at least $m+1$ anchors. In [1], we take this result one step further and show that in a sensor network where each sensor is only connected to a few neighboring sensors (that may not include any anchor) and only the distances with these neighbors are known at the sensors, the locations of any arbitrary number, $M \gg m$, of sensors can be resolved with at least $m+1$ anchors.

tion III discusses the problem formulation we adopt here. Section IV discusses network topology dynamics, whereas Sections V and VI consider the algorithm when more than $m+1$ anchors and more than $m+1$ neighbors are available, respectively. Section VII presents simulation studies and Section IX concludes the paper.

II. Background

In this section, we summarize the distributed localization algorithm in [1]. We introduce the following notation. In $\mathbb{R}^m$ $(m \geq 1)$, let Ω be the set of M sensors with unknown location and let κ be the set of $m+1$ anchors (that know their locations exactly) such that

$$\mathcal{C}(\Omega) \subset \mathcal{C}(\kappa), \tag{1}$$

where $\mathcal{C}(\cdot)$ denotes the convex hull formed by the elements of the set in its argument. The previous statement is another way of stating that the M sensors lie in the convex hull of $m+1$ anchors. Let Θ be the set of all of the nodes in the network, i.e., $\Theta = \Omega \cup \kappa$. We assume that for each $p \in \Omega$ there exists a triangulation set[2], $\Theta_p \subset \Theta$, such that

$$\begin{aligned} |\Theta_p| &= m+1 && (2)\\ p &\in \mathcal{C}(\Theta_p) && (3)\\ A_{\Theta_p} &> 0, && (4) \end{aligned}$$

where A_{Θ_p} is the generalized volume of the convex hull of Θ_p.

The location of a sensor, $p \in \Omega$, is now written as a convex combination of the locations of the elements in Θ_p

$$\mathbf{c}_p = \sum_{j \in \Theta_p} a_{pj} \mathbf{c}_j, \tag{5}$$

where $\mathbf{c}_p = [c_{1,p}, \ldots, c_{m,p}]$ is the (location) coordinate row of the pth sensor and a_{pj} are the barycentric coordinates [2], [3]. We have the following properties on a_{pj}.

$$\begin{aligned} a_{pj} &\in [0,1], \ \forall\ j, && (6)\\ \sum_{j \in \Theta_p} a_{pj} &= 1. && (7) \end{aligned}$$

The barycentric coordinates, a_{pj}, are computed by

$$a_{pj} = \frac{A_{\Theta_p \cup \{p\} - \{j\}}}{A_{\Theta_p}}. \tag{8}$$

It can be noted that, with the above equation for a_{pj}, the properties in (6) and (7) are satisfied. The barycentric coordinates, a_{pj}, are the ratios of the generalized volumes of two sets, see (8), and are computed using only the distance information among the elements in the set, $\Theta_p \cup \{p\}$, by using the Cayley-Menger determinants [4], see Appendix I.

[2]The identification of a triangulation set is an important step in the algorithm. A convex hull inclusion test is provided in [1] to test if a sensor lies in the convex hull of $m+1$ arbitrarily chosen sensors from its neighbors. Probabilistic bounds on the communication radius and the sensor deployment density are also provided in [1] to guarantee a successful triangulation with arbitrary high probability.

In (5), since the elements in Θ_p that lie in Ω also do not know their exact locations, we write the following iterative algorithm for the location of sensor p.

$$\mathbf{c}_p(t+1) = \begin{cases} \mathbf{c}_p(t), & p \in \kappa, \\ \sum_{j \in \Theta_p} a_{pj} \mathbf{c}_j(t), & p \in \Omega. \end{cases} \tag{9}$$

We now write the iterative algorithm given in (9) in matrix form. We have

$$\mathbf{C}(t+1) = \mathbf{Y}\mathbf{C}(t), \tag{10}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{m+1} \\ \mathbf{c}_{m+2} \\ \vdots \\ \mathbf{c}_n \end{bmatrix}, \tag{11}$$

and $n = m+1+M$. The first $m+1$ rows of $\mathbf{C}$ correspond to the anchors and hence keep their coordinates and the last M rows are updated according to (9). The iteration matrix, $\mathbf{Y}$, is partitioned as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix}, \tag{12}$$

where $\mathbf{I}_{m+1}$ is an $(m+1) \times (m+1)$ identity matrix, $\mathbf{B}$ is $M \times (m+1)$ matrix and $\mathbf{P}$ is an $M \times M$ matrix. Each row in the iteration matrix, $\mathbf{Y}$, sums to 1 and each element lies in the interval $[0,1]$. The iteration matrix, $\mathbf{Y}$, can be thought of as a transition probability matrix of an absorbing Markov chain and it can also be seen that the matrix $\mathbf{P}$ is a substochastic matrix, i.e., $\rho(\mathbf{P}) < 1$, where $\rho(\cdot)$ denotes the spectral norm of a matrix, see [1] for details.

A. Convergence

The iterative algorithm in (10) can be written as

$$\begin{aligned} \mathbf{C}(t+1) &= \mathbf{Y}^{t+1}\mathbf{C}(0), \\ &= \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \sum_{k=1}^{t} \mathbf{P}^k \mathbf{B} & \mathbf{P}^{t+1} \end{bmatrix} \mathbf{C}(0), \\ &\rightarrow \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B} & \mathbf{0} \end{bmatrix} \mathbf{C}(0), \end{aligned} \tag{13}$$

as t goes to infinity. The last equation follows from Lemma 4 provided in Appendix II. It can be noted that the iterative algorithm converges to a representation of each node in the network in terms of *only* the anchors. It is shown in [1] that this representation is the exact representation of the sensors with unknown locations in terms of the anchors, i.e.,

$$\mathbf{C}(t+1) \rightarrow \mathbf{C}^*, \tag{14}$$

where $\mathbf{C}^*$ is the exact coordinate matrix of the sensors plus the anchors. Furthermore, it can also be seen that the algorithm converges for any initial condition on the sensors in Ω. The proof is carried out using a coordinate transformation method [1]. In this paper, we give an alternate proof of the above iterative algorithm.

III. Problem Formulation

In this paper, we generalize our results in [1] to the following cases. (i) In [1], we required the iteration matrix, $\mathbf{Y}$, to be fixed in the algorithm. This assumption corresponds to static network topology. Here, we let the network topology to be dynamic, and hence a different iteration matrix can be chosen at each time step, t, of the algorithm. (ii) $|\kappa| > m+1$, i.e., the number of anchors can be greater than $m+1$. (iii) We also give a different proof (from [1]) for the convergence of the iterative algorithm. (iv) $|\Theta_p| > m+1$, i.e., the number of neighboring sensors that a particular sensor used to express its own coordinates can be more than $m+1$.

IV. Dynamic Network Topology

In case of dynamic network topology, each sensor, p, chooses a different neighborhood $\Theta_p(t)$ at each iteration t of the iterative algorithm, such that (2)–(4) holds for $\Theta_p(t)$. In this case, the coordinates of the pth sensor can be written as

$$\mathbf{c}_p(t+1) = \begin{cases} \mathbf{c}_p(t), & p \in \kappa, \\ \sum_{j\in\Theta_p(t)} a_{pj}(t)\mathbf{c}_j(t), & p \in \Omega. \end{cases} \tag{15}$$

Lemma 1: The localization algorithm (15) converges to the exact sensor locations, $\mathbf{C}^*$.

Proof: The resulting localization algorithm (15) is where the iteration matrix, $\mathbf{Y}$, becomes a function of time, $\mathbf{Y}(t)$, and can be written as

$$\begin{aligned} \mathbf{C}(t+1) &= \mathbf{Y}(t)\mathbf{C}(t), \\ &= \prod_{l=0}^{t} \mathbf{Y}(l)\mathbf{C}(0). \end{aligned} \tag{16}$$

Consider the matrices, $\mathbf{\Upsilon}_t$, given by the product in (16), i.e.,

$$\mathbf{\Upsilon}_t = \prod_{l=0}^{t} \mathbf{Y}(l) \tag{17}$$

where each $\mathbf{Y}(l)$ can be partitioned as shown in (12). It follows from the structure of the matrices, $\mathbf{Y}(l)$, that

$$\begin{aligned} \mathbf{\Upsilon}_t &= \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{B}_t & \mathbf{P}_t \end{bmatrix} \cdots \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{B}_0 & \mathbf{P}_0 \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{J}_t & \prod_{l=0}^{t} \mathbf{P}_l \end{bmatrix}. \end{aligned} \tag{18}$$

Let the exact coordinate matrix, $\mathbf{C}^*$, be partitioned into the exact coordinates of the anchors, $\mathbf{C}^*_\kappa$, and the exact coordinates of the sensors, $\mathbf{C}^*_\Omega$, as

$$\mathbf{C}^* = \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}^*_\Omega \end{bmatrix}. \tag{19}$$

Since, $\mathbf{C}^*$ is the fixed point of each $\mathbf{Y}(l)$ in the product matrix, $\mathbf{\Upsilon}_t$, i.e.,

$$\begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}^*_\Omega \end{bmatrix} = \mathbf{Y}(l) \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}^*_\Omega \end{bmatrix}, \qquad \forall l, \tag{20}$$

it follows that $\mathbf{C}^*$ is the fixed point of the product matrix, $\mathbf{\Upsilon}_t$, in (17).

In particular,

$$\mathbf{J}_t\mathbf{C}^*_\kappa + \left(\prod_{l=0}^{t} \mathbf{P}_l\right)\mathbf{C}^*_\Omega = \mathbf{C}^*_\Omega, \qquad \forall t. \tag{21}$$

Since the matrix $\mathbf{P}_l$ is substochastic for any $l = 0, \ldots, t$ (see [1]), we have

$$\lim_{t\to\infty} \prod_{l=0}^{t} \mathbf{P}_l = 0. \tag{22}$$

Therefore, it follows from (21), that $\lim_{t\to\infty} \mathbf{J}_t\mathbf{C}^*_\kappa$ exists and, in particular,

$$\lim_{t\to\infty} \mathbf{J}_t\mathbf{C}^*_\kappa = \mathbf{C}^*_\Omega. \tag{23}$$

Now let

$$\mathbf{C}(0) = \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}_\Omega \end{bmatrix}, \tag{24}$$

be the actual initial state of the algorithm where $\mathbf{C}_\Omega$ is any arbitrary initial guess of the sensor locations. Then,

$$\begin{aligned} \lim_{t\to\infty} \mathbf{C}(t) &= \lim_{t\to\infty} \mathbf{\Upsilon}_t \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}_\Omega \end{bmatrix}, \\ &= \lim_{t\to\infty} \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{J}_t\mathbf{C}^*_\kappa + \left(\prod_{l=0}^{t} \mathbf{P}_l\right)\mathbf{C}_\Omega \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{C}^*_\kappa, \\ \mathbf{C}^*_\Omega \end{bmatrix}. \end{aligned} \tag{25}$$

■

V. More than $m+1$ anchors

In this section, we study the case where the number of anchors is greater than $m+1$, i.e., $|\kappa| = K > m+1$. This happens when, for instance, M sensors do not lie in the convex hull of $m+1$ anchors, but, lie in the convex hull of $K > m+1$ anchors. The iterative procedure has the same form as (10), however, the total number of sensors plus anchors becomes $K+M$. The coordinate matrix, $\mathbf{C}$, has the dimension $(K+M)\times m$ and the iteration matrix, $\mathbf{Y}_K$, has the dimension $(K+M)\times(K+M)$ that can be partitioned as

$$\mathbf{Y}_K = \begin{bmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix}, \tag{26}$$

where $\mathbf{I}_K$ is a $K\times K$ identity matrix, $\mathbf{B}$ is an $M\times K$ matrix and $\mathbf{P}$ is an $M\times M$ matrix.

Lemma 2: The iterative localization algorithm with $K > m+1$ anchors resulting into the iteration matrix, $\mathbf{Y}_K$, in (26) converges to the exact sensor locations, $\mathbf{C}^*$.

Proof: Since $\mathbf{C}^*$ is the fixed point of the (10), we have

$$\begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}^*_\Omega \end{bmatrix} = \begin{bmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{C}^*_\kappa \\ \mathbf{C}^*_\Omega \end{bmatrix}, \tag{27}$$

$$\Rightarrow \mathbf{C}^*_\Omega = \mathbf{B}\mathbf{C}^*_\kappa + \mathbf{P}\mathbf{C}^*_\Omega, \tag{28}$$

which gives

$$\mathbf{C}^*_\Omega = (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B}\mathbf{C}^*_\kappa. \tag{29}$$

Since $\mathbf{P}$ is a substochastic matrix with $\rho(\mathbf{P}) < 1$, the eigenvalues of $\mathbf{P}$ lie in $[0, 1)$, the eigenvalues of $\mathbf{I}_M - \mathbf{P}$ lie in $(0, 1]$ and hence $\mathbf{I}_M - \mathbf{P}$ is invertible.

Using Lemma 4 again, it can be shown that

$$\lim_{t\to\infty} \mathbf{Y}_K^{t+1} = \begin{bmatrix} \mathbf{I}_K & \mathbf{0} \\ (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B} & \mathbf{0} \end{bmatrix}. \quad (30)$$

Hence the iterative algorithm converges to

$$\lim_{t\to\infty} \mathbf{C}^{t+1} = (\mathbf{I}_M - \mathbf{P})^{-1}\mathbf{B}\mathbf{C}_\kappa^* = \mathbf{C}_\Omega^*. \quad (31)$$

This completes the proof for the case when we have more than $m+1$ anchors. ■

The proof for $K = m+1$ can be formulated as a special case of the above arguments and, hence, the above argument provides an alternative proof for the localization algorithm presented in [1].

VI. More than $m+1$ neighbors

Motivated by wireless sensor networks (WSNs), where each sensor broadcasts its data in a communication radius and every other sensor that lies in its communication radius can receive its data, we consider the case when a sensor can have more than $m+1$ neighboring nodes. Let $\widehat{\Theta}_p$ denote the set of sensors or anchors that lie in the communication radius, R_p, of sensor p, i.e.,

$$\widehat{\Theta}_p = \{j : d_{pj} < R_p\}, \quad (32)$$

where d_{pj} is the Euclidean distance between node p and node j. Let $\overline{\Theta}_p = \{\Theta_p^i\} \subseteq \widehat{\Theta}_p$ be the collection of subsets of $\widehat{\Theta}_p$ such that for each element, $\Theta_p^i \in \overline{\Theta}_p$, (2)–(4) holds. If $\overline{\Theta}_p = \varnothing$, the pth sensor increases its communication radius, R_p, until $|\overline{\Theta}_p| \geq 1$ (note that $|\varnothing| = 0$). In this fashion, each sensor can adaptively choose its communication radius, R_p, large enough such that $|\overline{\Theta}_p| \geq 1$.

The coordinates of sensor p's location can now be expressed uniquely in terms of any element, $\Theta_p^i \in \overline{\Theta}_p$, by using (5). Furthermore, each sensor p can express its coordinates in terms of all the elements in $\overline{\Theta}_p$ as a convex combination of each of them, i.e.,

$$\mathbf{c}_p = \sum_i w_p^i \sum_{j\in\Theta_p^i} a_{pj}^i \mathbf{c}_j, \qquad p \in \Omega, \quad (33)$$

where $w_p^i \geq 0\ \forall i$ and $\sum_i w_p^i = 1$. An iterative procedure obtained on (33) is given by

$$\mathbf{c}_p(t+1) = \begin{cases} \mathbf{c}_p(t), & p \in \kappa, \\ \sum_i w_p^i \sum_{j\in\Theta_p^i} a_{pj}^i \mathbf{c}_j(t), & p \in \Omega. \end{cases} \quad (34)$$

Lemma 3: The distributed localization algorithm in (34) converges to the exact sensor locations, $\mathbf{C}^*$.

Proof: The distributed localization algorithm in (34) can be written in matrix form as

$$\mathbf{C}(t+1) = \widetilde{\mathbf{Y}}\mathbf{C}(t). \quad (35)$$

As shown in (33), we again note that the way we have derived the iteration matrix, $\widetilde{\mathbf{Y}}$, the matrix of exact coordinates, $\mathbf{C}^*$, still remains the fixed point of the algorithm, i.e.,

$$\begin{bmatrix} \mathbf{C}_\kappa^* \\ \mathbf{C}_\Omega^* \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \widetilde{\mathbf{B}} & \widetilde{\mathbf{P}} \end{bmatrix} \begin{bmatrix} \mathbf{C}_\kappa^* \\ \mathbf{C}_\Omega^* \end{bmatrix}, \quad (36)$$

$$\Rightarrow \mathbf{C}_\Omega^* = \widetilde{\mathbf{B}}\mathbf{C}_\kappa^* + \widetilde{\mathbf{P}}\mathbf{C}_\Omega^*, \quad (37)$$

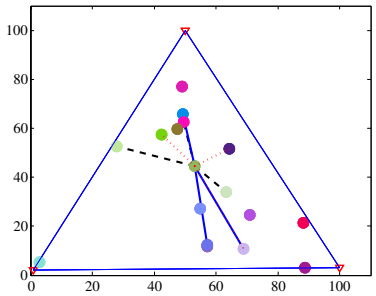


Fig. 1. For a fixed sensor, its 3 different neighborhoods are shown.

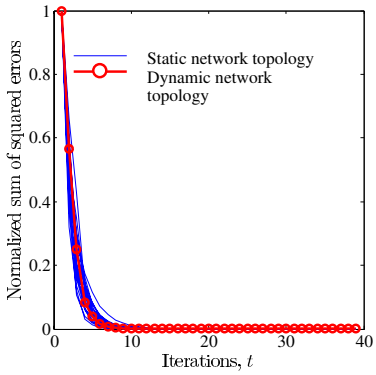


Fig. 2. Performance comparison of the dynamic scheme with $T = 20$ static (fixed) topologies.

which gives

$$\mathbf{C}_\Omega^* = \left(\mathbf{I}_M - \widetilde{\mathbf{P}}\right)^{-1} \widetilde{\mathbf{B}}\mathbf{C}_\kappa^*. \quad (38)$$

It is straightforward to show that $\widetilde{\mathbf{P}}$ is a substochastic matrix since it is a convex combination of substochastic matrices.

Using Lemma 4 again, it can be shown that

$$\lim_{t\to\infty} \widetilde{\mathbf{Y}}^{t+1} = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ (\mathbf{I}_M - \widetilde{\mathbf{P}})^{-1}\widetilde{\mathbf{B}} & \mathbf{0} \end{bmatrix}. \quad (39)$$

Hence the iterative algorithm converges to

$$\lim_{t\to\infty} \mathbf{C}^{t+1} = (\mathbf{I}_M - \widetilde{\mathbf{P}})^{-1}\widetilde{\mathbf{B}}\mathbf{C}_\kappa^* = \mathbf{C}_\Omega^*. \quad (40)$$

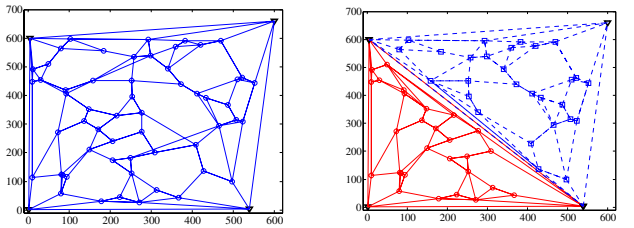
■

VII. Simulations

In this section, we present numerical experiments.

A. Case 1: Dynamic Network Topology

We simulate an $n = 20$ node network in $m = 2-$Euclidean space, where we have $K = m+1 = 3$ anchors (with known locations) and $M = 17$ sensors (with unknown locations). We formulate $T = 20$ different iteration matrices, $\mathbf{Y}(l)$, where $l = 1, \ldots, 20$ and at each iteration, t, of the algorithm we randomly choose one out of the 20 iteration matrices. The dynamical network topology for a particular sensor is shown in Fig. 1 where 3 different neighborhoods are shown. Fig. 2 compares the performance of the dynamic scheme with $T = 20$ static networks.



(a) (b)

Fig. 3. (a) The overall sensor network with $K = 4 > m+1$ anchors such that $\mathcal{C}(\Omega) \subset \mathcal{C}(\kappa)$. (b) Dividing the overall network into two subproblems where we have $m + 1 = 3$ anchors for each of the subproblems.

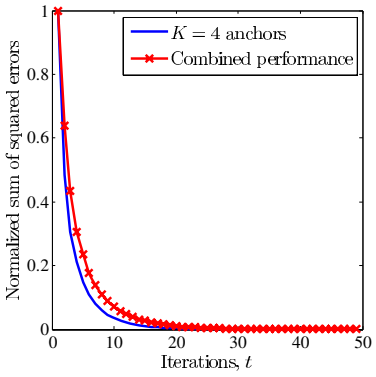


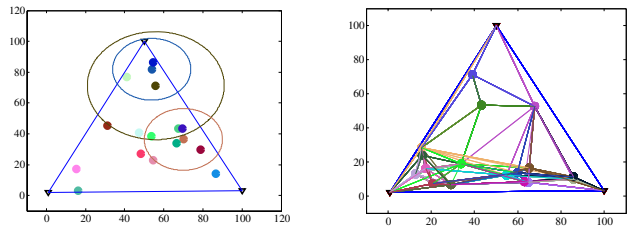
Fig. 4. Performance comparison between the aggregated performance of the two subproblems and the scheme with $K = 4$ anchors.

B. Case 2: More than $m + 1$ anchors

We simulate an $n = 60$ node network in $m = 2-$Euclidean space, where we have $K = 4 > m + 1$ anchors (with known locations) and $M = 56$ sensors (with unknown locations). Fig. 3(a) shows the overall network where the $M = 56$ sensors lie in the convex hull of $K = 4$ anchors. Fig. 3(b) divides the original problem into two subproblems, each of which is solving the unknown sensors with $m + 1 = 3$ anchors. Fig. 4 compares the combined performance of the two subproblems with the performance of the scheme where we used more anchors.

C. Case 3: More than $m + 1$ neighbors

We simulate an $n = 20$ node network in $m = 2-$Euclidean space, where we have $K = m + 1 = 3$ anchors (with known locations) and $M = 17$ sensors (with unknown locations). The neighborhood, Θ_p^i, of the pth sensor is chosen adaptively by increasing the communication radius, R_p, of the pth sensor as discussed in Section VI. Adaptive choice of the communication radius is shown in Fig. 5(a) for three arbitrarily chosen sensors. Fig. 5(b) shows the resulting communication network where it can be verified that each sensor is now connected to more than $m + 1$ neighbors. Performance comparison of fixed $m+1$ neighbors for each $T = 20$ different neighborhoods with the scheme where all the neighborhoods are combined using a weighting



(a) (b)

Fig. 5. (a) Adaptively choosing the communication radius, R_p shown for three arbitrarily chosen sensors. (b) Resulting network where each sensor is connected to more than $m + 1 = 3$ neighbors.

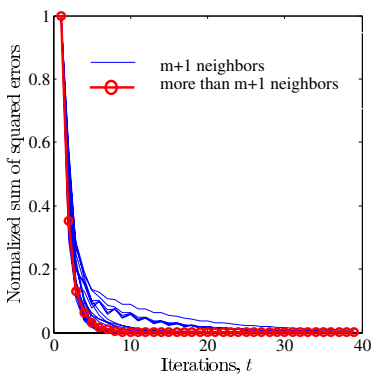


Fig. 6. Performance of the fixed $m + 1$ neighbors with more than $m + 1$ neighbors.

sequence is shown in Fig. 6. The combining weights are chosen to be $1/N_p$, where $N_p = |\overline{\Theta_p}|$.

VIII. Remarks

It is a straightforward generalization to combine all of the three scenarios presented in this paper. The resulting algorithm gives a comprehensive distributed localization algorithm that deals with random network topologies, any number, $K \geq m + 1$, of anchors and incorporates all the sensors in the neighborhood of each sensor to achieve sensor localization. In practical wireless sensor network applications, the sensing environment is, in general, random, leading to communication link failures among the sensors, randomness in the system parameters, and quantized data exchange due to bandwidth restrictions. In other work, we have considered noisy and quantized communications in consensus, with possibly random link failures, [27], [28] using stochastic approximation to prove almost sure convergence. Similar generalizations can be considered for the set-up presented here. Under broad assumptions of environment uncertainty, the distributed localization algorithm can be extended to account for these random phenomena, see [1], which treats the scenario when noisy inter-sensor distance measurements are available and inter-sensor communication is imperfect.

IX. Conclusions

In this paper, we extended our work on distributed sensor localization in [1] and presented a comprehensive generalization to our approach. This contribution provides a complete framework to apply the localization algorithm to existing wireless sensor applications. The convergence results are proved in all cases and the convergence is shown to be exact. In particular, we provide the following results: (i) Choosing the sensor network communication topology dynamically improves the worst case performance; (ii) Increasing the number of anchors increases the convergence of the algorithm. (iii) Increasing the number of neighbors significantly improves the worst case performance. We further provide extensive simulations to support the theoretical claims.

Appendix I
Cayley-Menger Determinant

The Cayley-Menger determinant provides the generalized volume, A_κ, of $\mathcal{C}(\kappa)$ [4]. Let $\mathbf{1}_{m+1}$ denote a column vector of $m+1$ 1s, the Cayley-Menger determinant is given by

$$A_\kappa^2 = \frac{1}{s_{m+1}} \begin{vmatrix} 0 & \mathbf{1}_{m+1}^T \\ \mathbf{1}_{m+1} & \mathbf{\Gamma} \end{vmatrix}, \tag{41}$$

where $\mathbf{\Gamma} = \{d_{lj}^2\}, l, j \in \kappa$, is the matrix of squared distances, d_{lj}, among the $m+1$ points in κ and

$$s_m = \frac{2^m (m!)^2}{(-1)^{m+1}}, \qquad m = \{0, 1, 2, \ldots\}. \tag{42}$$

Appendix II
A Matrix Result

Lemma 4: If a matrix, $\mathbf{P}$, is such that

$$\rho(\mathbf{P}) < 1, \tag{43}$$

where $\rho(\cdot)$ denotes the spectral norm, then

$$\lim_{t\to\infty} \mathbf{P}^{t+1} = \mathbf{0}, \tag{44}$$

$$\lim_{t\to\infty} \sum_{k=0}^{t} \mathbf{P}^k \mathbf{B} = (\mathbf{I}_M - \mathbf{P})^{-1} \mathbf{B}. \tag{45}$$

Proof: See [1]. ∎

References

[1] Usman A. Khan, Soummya Kar, and José M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," submitted to IEEE Trans. on Signal Processing, http://arxiv.org/PS_cache/arxiv/pdf/0802/0802.3563v2.pdf, Feb. 2008.

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

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

[4] 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.

[5] N. Patwari, A. O. Hero III, J. Ash, R. L. Moses, S. Kyperountas, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[6] D. Niculescu and B. Nath, "Ad hoc positioning system," in *IEEE Globecom*, Apr. 2001, pp. 2926–2931.

[7] 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.

[8] 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.

[9] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *IEEE ICASSP*, Salt Lake City, UA, May 2001, pp. 2037–2040.

[10] 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.

[11] 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.

[12] 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.

[13] Usman Khan, Soummya Kar, and José M. F. Mour, "Distributed algorithms in sensor networks," in *Handbook on sensor and array processing*, Simon Haykin and K. J. Ray Liu, Eds. Wily-Interscience, New York, NY, 2009, to appear, 33 pages.

[14] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computations*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[15] G. V. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal on Parallel and Distributed Computing*, vol. 7, pp. 279–301, 1989.

[16] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[17] S. Kar, S. A. Aldosari, and J. M. F. Moura, "Topology for distributed inference on graphs," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2609–2613, June 2008.

[18] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, July 2008.

[19] R. Olfati-Saber, J. Alex Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *IEEE Proceedings*, vol. 95, no. 1, pp. 215–233, January 2007.

[20] R. R. Tenney and N. R. Sandell, "Detection with distributed sensors," vol. AES-17, no. 4, pp. 501–510, July 1981.

[21] J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," vol. AC-30, no. 5, pp. 440–446, May 1985.

[22] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," vol. AC-31, no. 9, pp. 803–812, September 1986.

[23] J. N. Tsitsiklis, "Decentralized detection," in *Advances in Statistical Signal Processing*, H. V. Poor and J. B. Thomas, Eds., pp. 297–344. JAI Press, Greenwich, CT, USA, 1993.

[24] Saeed Aldosari and José M. F. Moura, "Fusion in sensor networks with communication constraints," in *IPSN04 Information Processing in Sensor Networks*, Berkeley CA, April 2004, IEEE/ACM, pp. 108–115.

[25] Saeed Aldosari and José M. F. Moura, "Detection in decentralized sensor networks," in *ICASSP04, IEEE International Conference on Signal Processing*, Montreal, Québec, Canada, May 17-21 2004, IEEE, vol. II, pp. 277–280.

[26] S. Aldosari and J. M. F. Moura, "Detection in sensor networks: The saddlepoint approximation," *IEEE Transactions on Signal Processing*, vol. 55, no. 1, pp. 327–340, January 2007.

[27] S. Kar and José M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Transactions on Signal Processing*, 2008, Accepted for publication, 30 pages.

[28] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data," Submitted for publication, 30 pages, see http://arxiv.org/abs/0712.1609, November 2007.