A distributed range-based algorithm for localization in mobile networks

Sam Safavi, *Student Member, IEEE*, and Usman A. Khan[†], *Senior Member, IEEE*

***Abstract*—In this paper, we provide a distributed algorithm to locate an arbitrary number of agents moving in a bounded region. Assuming that each agent can estimate a noisy version of its motion and the distances to the nodes in its communication radius, we provide a simple *linear* update to find the locations of an arbitrary number of mobile agents when they follow some convexity in their deployment and motion, given *at least one* anchor, agent with known location, is present in $\mathbb{R}^m$. At each iteration, agents update their location estimates as a convex combination of the states of the neighbors, if they lie inside their convex hull, and do not update otherwise. We abstract the corresponding localization algorithm as a Linear Time-Varying (LTV) system, and using *slice* notation we show that it asymptotically converges to the true locations of the agents. We study the effects of noise on our localization algorithm, and provide simulations to verify our analytical results.**

I. Introduction

Localization refers to a collection of algorithms that estimate the location of nodes in a network. Relevant applications include robotics, traffic control, and environment monitoring. Localization algorithms can be broadly classified into centralized and distributed approaches. Despite the benefits of centralized algorithms such as higher accuracy, they are impractical in large networks due to the limited power and communication capabilities. On the other hand, distributed localization techniques such as successive refinements, [1], [2], probabilistic approaches, [3], multilateration, [4], [5], and graph theoretical methods, [6], [7] are considered to be more efficient and easier to implement in large scale networks. In particular, in [8] a distributed localization (DILOC) algorithm based on the barycentric coordinates is provided. DILOC requires all agents to lie inside the convex hull of (at least) $m+1$ anchors in $\mathbb{R}^m, m \geq 1$, i.e., to satisfy *global convexity*. Since all agents may not be able to communicate with the anchors, DILOC further assumes that each agent has $m+1$ neighbors such that it lies inside their convex hull, i.e., each agent satisfies *local convexity*. All agents iteratively update their location estimates as a linear-convex combination of such $m+1$ neighbors, and DILOC converges to the true agent locations if local and global convexity assumptions are satisfied at each agent. Indeed, DILOC requires somewhat stringent deployment implied by satisfying the two convexity conditions. In this context, Refs. [9], [10] extend DILOC by providing a barycentric representation that does not require convexity. The fundamental notion behind these barycentric-based methods, [8]–[11], is that the agents implement linear (not linearized) iterations that are guaranteed to converge regardless of the initial conditions, unlike the nonlinear localization algorithms. In particular, DILOC can be abstracted as a Linear Time-Invariant (LTI) system, which is stable under the convexity conditions and thus forgets the initial conditions.

A related algorithm, [12], is developed for mobile agents, which assumes that mobile agents satisfy both convexity conditions *at each time instant*. However, in a mobile network the agents may move inside and outside of the convex hull, and it is highly unlikely that a network of mobile agents, [13], [14], will follow any (local and/or global) convexity at all times. Other localization algorithms specifically designed for mobile networks have been proposed in [15]–[18], most of which use Sequential Monte Carlo (SMC). Despite the simplicity in implementation, SMC methods are time-consuming as they need to continue sampling and filtering until enough samples are obtained to represent the posterior distribution of a mobile agent's position.

In this paper, we consider localization of a mobile network, assuming that each agent knows a noisy version of its motion and its distances to the neighboring nodes (agents and/or anchors). Similar to DILOC, we are interested in implementing linear-convex, distributed iterations that converge to the true locations regardless of the agents' initial conditions. However, the implementation and the stability analysis is more challenging for a mobile network, because: (i) the agents may move inside and outside of the convex hull formed by the anchors, hence *global convexity* may not be satisfied; (ii) *local convexity* may not be satisfied *at all times*; (iii) the neighborhood at each agent is dynamic, which results into a Linear Time-Varying (LTV) system. As we will discuss in Section III, the resulting LTV system comprises of system matrices that may be *identity*, when no agent is able to satisfy local convexity; *stochastic*, when an agent finds $m+1$ neighboring *agents* passing local convexity; or *strictly sub-stochastic*, when this neighborhood includes at least one anchor. As a result, establishing the asymptotic behavior of such an LTV system is non-trivial.

In order to deal with these issues, we propose a localization algorithm for mobile networks, which can be described as follows. Agent i needs $m+1$ nodes that satisfy local convexity to *linearly* update its location estimate. At each iteration, agent i uses the distances to test for local convexity, this test is explained in Section II-A. If the test fails, the agent continues to travel in the network and meets new nodes. Eventually, the local convexity is satisfied and agent i updates its location with respect to the nodes in the convex hull. Using this approach, we show that agent locations are refined as the procedure continues and the algorithm converges to the true agent locations. To this aim, we partition the entire chain

[†]The authors are with the Department of Electrical and Computer Engineering, Tufts University, 161 College Ave, Medford, MA 02155, {sam.safavi@,khan@ece.}tufts.edu. This work is partially supported by an NSF Career award: CCF # 1350264.

of system matrices into non-overlapping slices, and relate the convergence of the network to the lengths of these slices.

The rest of this paper is organized as follows. In Section II, we formulate the problem. We propose our localization algorithm that does not require either local or global convexity in Section III. We then provide the convergence analysis in Section IV, and study the effects of noise in Section V. We present simulation results in Section VI, and finally, Section VII concludes the paper.

II. Preliminaries and Problem formulation

Consider a network of N mobile agents with unknown locations, in the set Ω, and M anchors with known locations, in the set κ, all located in $\mathbb{R}^m$, $m \geq 1$. Note that in most localization algorithms $m+1$ anchors are required to localize an agent with unknown location in $\mathbb{R}^m$, $m \geq 1$. Let $\Theta = \Omega \cup \kappa$ be the set of all nodes in the network, and $\mathbf{x}_k^{i*} \in \mathbb{R}^m$ be an m-dimensional row vector that denotes the *true location* of the i-th node, $i \in \Theta$, at time k, where $k \geq 0$ is the discrete-time index. We assume that each agent is able to measure its distance to the nearby nodes by using RSSI, ToA, TDoA, or camera-based techniques, [19], [20]. The problem is to estimate the locations of the mobile agents in the set Ω. We now describe DILOC, which was originally introduced in [8].

A. DILOC

DILOC is a distributed localization algorithm, which considers static networks, i.e., $\mathbf{x}_k^{i*} = \mathbf{x}^{i*}, \forall i \in \Theta$. It assumes that each agent satisfies *global convexity*, i.e., it lies inside the convex hull of $m+1$ anchors, denoted by $\mathcal{C}(\kappa)$. It further assumes that each agent, say i, satisfies *local convexity*, i.e., lies in the convex hull of a set of $m+1$ neighbors, referred to as a *triangulation set*, Θ_i. Using only inter-node distances in any arbitrary dimension, m, a convex hull inclusion test is

$$i \in \mathcal{C}(\Theta_i), \qquad \text{if} \sum_{j \in \Theta_i} A_{\Theta_i \cup \{i\} \setminus j} = A_{\Theta_i}, \tag{1}$$

in which A_{Θ_i} denotes the m-dimensional volume of $\mathcal{C}(\Theta_i)$, and can be computed by using the Cayley-Menger determinant, [8], [21], see Fig. 1.

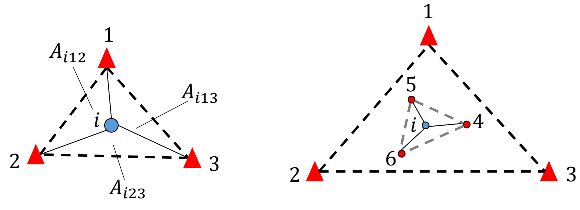

Fig. 1. $\mathbb{R}^2$: (Left) Global convexity. (Right) Local convexity: Agent i lies inside the triangulation set formed by agents, 4, 5 and 6; Red triangles represent anchors.

The i-th sensor updates its location estimate, $\mathbf{x}_k^i$ at time k, according to the following *linear-convex* combination:

$$\mathbf{x}_{k+1}^i = \sum_{j \in \Theta_i \cap \Omega} \frac{A_{\Theta_i \cup \{i\} \setminus j}}{A_{\Theta_i}} \mathbf{x}_k^j + \sum_{j \in \Theta_i \cap \kappa} \frac{A_{\Theta_i \cup \{i\} \setminus j}}{A_{\Theta_i}} \mathbf{x}^{j*}, \tag{2}$$

where Θ_i is the triangulation set for sensor $i \in \Omega$, and the coefficients, $\frac{A_{\Theta_i \cup \{i\} \setminus j}}{A_{\Theta_i}}$, are the *barycentric coordinates* associated to Möbius, [22]. Assuming non-trivial configurations, i.e., $A_\kappa \neq 0, A_{\Theta_i} \neq 0, \forall i \in \Omega$, it is shown in [8] that for any initial conditions DILOC converges to the true sensor locations if each sensor successfully finds a triangulation set.

B. Assumptions

In this paper, we adapt DILOC to deal with the challenges in mobile networks, where the global and local convexity may not be satisfied at all times. To this aim, we assume that the true locations of the nodes evolve according to the following random motion model

$$\mathbf{x}_{k+1}^{i*} = \mathbf{x}_k^{i*} + \widetilde{\mathbf{x}}_{k+1}^i, \qquad i \in \Theta, \tag{3}$$

where $\widetilde{\mathbf{x}}_k^i$ is the true motion vector at time k such that each node, $i \in \Theta$, remains in a bounded region of interest in $\mathbb{R}^m$. We assume that agent i measures a noisy version, $\widehat{\mathbf{x}}_k^i$, of this motion:

$$\widehat{\mathbf{x}}_k^i = \widetilde{\mathbf{x}}_k^i + n_k^i, \tag{4}$$

in which n_k^i represents the measurement noise at time k. We denote the distance between any two nodes, i and j, measured at the time of communication, k, by $\widetilde{d}_k^{ij}$, and we assume that the distance measurement, $\widehat{d}_k^{ij}$, at agent i is not perfect and includes noise, i.e.,

$$\widehat{d}_k^{ij} = \widetilde{d}_k^{ij} + r_k^{ij}, \tag{5}$$

where r_k^{ij} is the noise in the distance measurement at time k.

We now enlist our assumptions:

A0: Anchor locations, $\mathbf{x}_k^{i*}, i \in \kappa$, are perfectly known at all times.

A1: Each agent, $i \in \Omega$, knows a noisy version, $\widehat{\mathbf{x}}_k^i$, of its motion, $\widetilde{\mathbf{x}}_k^i$, at all times, see Eq. (4).

A2: Each agent, $i \in \Omega$, obtains a noisy measurement, $\widehat{d}_k^{ij}$, to every node, $j \in \Theta$, in its communication radius, r, at time k, see Eq. (5).

Under the above assumptions, which are common in the related literature, e.g., see [16], [18], we are interested in finding the true locations of each agent without the presence of any central coordinator[1].

In what follows, we first assume that the motion and distance measurements are not effected by noise, i.e., $n_k^i = 0$ and $r_k^{ij} = 0$, in Eqs. (4) and (5). We then examine the impact of noise in Section V, and provide modifications to the algorithm to deal with the undesirable effects of noise.

III. Distributed Localization Algorithm

Consider a network of N mobile agents and M anchors in $\mathbb{R}^m$. Let $\mathcal{N}_i(k) \subseteq \Theta$ be the set of neighbors of agent, $i \in \Omega$, at time k. We now describe the localization algorithm. Each agent starts with a random guess of its location estimate. At each time $k > 0$ there are two different update scenarios for any arbitrary agent, i: If $0 \leq |\mathcal{N}_i(k)| < m+1$, agent i does

[1] Although our approach is applicable to arbitrary dimensions, $m > 2$, we use $\mathbb{R}^2$ in the remainder of the paper for simplicity and ease of notation.

not update its current location estimate. On the other hand, if $|\mathcal{N}_i(k)| \geq m+1$, agent i performs the inclusion test. If the test is passed agent i applies the following location update:

$$\mathbf{x}_{k+1}^i = \alpha_k \mathbf{x}_k^i + (1-\alpha_k) \sum_{j\in\Theta_i(k)} a_{ij}(k)\mathbf{x}_k^j + \widetilde{\mathbf{x}}_{k+1}^i, \tag{6}$$

in which $\Theta_i(k)$ is the triangulation set at time k, $a_{ij}(k)$ is the barycentric coordinate of node i with respect to the nodes $j \in \Theta_i(k)$, and α_k is a design parameter such that

$$\alpha_k = \left\{ \begin{array}{ll} 1, & \forall k \mid \Theta_i(k) = \emptyset, \\ \in [\beta, 1)\,, & \forall k \mid \Theta_i(k) \neq \emptyset. \end{array} \right. \tag{7}$$

While $N_i(k) = \emptyset$ implies that node i has no neighbors at time k, note that $\Theta_i(k) = \emptyset$ implies that at time k no set of neighbors meet the local convexity condition, i.e., agent i can not find any triangulation set among the neighbors. By separating the barycentric coordinates corresponding to the neighboring agents and anchors, we can rewrite Eq. (6) as

$$\begin{aligned} \mathbf{x}_{k+1}^i &= \alpha_k \mathbf{x}_k^i + (1-\alpha_k)\left(\sum_{j\in\Theta_i(k)\cap\Omega} p_{ij}(k)\mathbf{x}_k^j\right), \\ &+ (1-\alpha_k)\left(\sum_{m\in\Theta_i(k)\cap\kappa} b_{im}(k)\mathbf{u}_k^m\right) + \widetilde{\mathbf{x}}_{k+1}^i, \end{aligned} \tag{8}$$

such that

$$a_{ij} = \begin{cases} p_{ij}, & \text{if } j \in \Theta_i(k)\cap\Omega, \\ b_{im}, & \text{if } m \in \Theta_i(k)\cap\kappa. \end{cases} \tag{9}$$

It can be inferred from Eq. (7) that the self-weight at each agent is always lower bounded, i.e.,

$$0 < \beta \leq p_{ii}(k) \leq 1, \forall k, i \in \Omega. \tag{10}$$

The above algorithm can be expressed in matrix form as

$$\mathbf{x}_{k+1} = \mathbf{P}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \widetilde{\mathbf{x}}_{k+1}, \qquad k > 0, \tag{11}$$

where $\mathbf{x}_k$ is the vector of agent coordinates at time k, $\mathbf{u}_k$ is the vector of anchor coordinates at time k, and $\widetilde{\mathbf{x}}_{k+1}$ is the change in the location of agents at the beginning of the k-th iteration according to the motion model. Also $\mathbf{P}_k$, and $\mathbf{B}_k$, the system matrix and the input matrix of the above LTV system, contain the barycentric coordinates with respect to the agents with unknown locations, and anchors, respectively. Since anchors play the role of the input in the above LTV system and inject accurate information into the network, it is reasonable to set a lower bound on the weights assigned to the anchor states. To this aim, we make an additional assumption as follows.

A3: Anchor contribution. If an anchor is involved in an update, i.e., for any $b_{im}(k) \neq 0$, we assume that

$$0 < \alpha \leq b_{im}(k), \qquad \forall k, i \in \Omega, m \in \Theta_i(k)\cap\kappa. \tag{12}$$

Equivalently, we have

$$0 < \alpha \leq (\mathbf{B}_k)_{ij}, \qquad \forall j \in \Theta_i(k)\cap\kappa, \tag{13}$$

where $(\mathbf{B}_k)_{ij}$ is the (i,j)-th element of the input matrix, $\mathbf{B}_k$.

Note that with the lower bounds on both the self-weights and the weights assigned to the anchors, according to Eqs. (10) and (13), the matrix of barycentric coordinates with respect to agents with unknown locations, i.e., the system matrix at time k, $\mathbf{P}_k$, is either (i) *identity*, when no update occurs; or, (ii) *identity except a stochastic i-th row*, when there is no anchor in the triangulation set at time k, i.e., $\Theta_i(k)\cap\kappa = \emptyset$; or, (iii) *identity except a strictly sub-stochastic i-th row*, when there is at least one anchor in the triangulation set at time k.

In the next section, we provide sufficient conditions for the iterative localization algorithm, Eq. (11), to converge to the true agent locations. Before we proceed, let us make the following definitions:

Definition 1. *A non-negative, stochastic matrix is such that all of its rows sum to one.*

Definition 2. *A non-negative, strictly sub-stochastic matrix is such that it has at least one row that sums to strictly less than one and every other row sums to at most one.*

IV. Distributed Mobile Localization: Analysis

We now provide a related result on the convergence of an infinite product of (sub-) stochastic matrices, [23], that we will use to study the convergence of Eq. (11).

A. Asymptotic stability of stochastic LTV systems

Consider an LTV system: $\mathbf{x}_{k+1} = \mathbf{P}_k\mathbf{x}_k$, such that the time-varying system matrix, $\mathbf{P}_k$, is random, and represents at most one state update, say the i-th state, for any k, i.e., at most one row, i, of $\mathbf{P}_k$ is different from identity. In addition, assume the following on the update at time k:

B0: If the updating row, i, in $\mathbf{P}_k$ is stochastic,

$$0 < \beta_1 \leq (\mathbf{P}_k)_{i,i}, \qquad \beta_1 \in \mathbb{R}, \tag{14}$$

B1: If the updating row, i, in $\mathbf{P}_k$ is strictly sub-stochastic,

$$\sum_j (P_k)_{i,j} \leq \beta_2 < 1, \qquad \beta_2 \in \mathbb{R}. \tag{15}$$

To investigate the asymptotic behavior of an LTV system with such system matrices, we introduce the notion of a slice, M_j, as the smallest product of consecutive system matrices, such that each slice has a subunit infinity norm, i.e., $\|M_t\|_\infty < 1, \forall t$, *and* the entire set of systems matrices is covered by non-overlapping slices, i.e., $\prod_t M_t = \prod_k P_k$. Each slice is initiated by a strictly sub-stochastic system matrix, and terminated after all rows become strictly sub-stochastic. The length of a slice is defined as the number of matrices forming the slice, and the upper bound on the infinity norm of a slice is further related to the length of the slice [23]. Asymptotic behavior of the above LTV system is characterized in the following theorem.

Theorem 1. *With Assumptions* **B0-B1***, the LTV system, $\mathbf{x}_{k+1} = \mathbf{P}_k\mathbf{x}_k$, converges to zero if for every $i \in \mathbb{N}$, there exists a set, J_1, of slices such that*

$$\exists M_j \in J_1 : \quad |M_j| \leq \frac{1}{\ln(\beta_1)} \ln\left(\frac{1 - e^{(-\gamma_2 i^{-\gamma_1})}}{1-\beta_2}\right) + 1, \tag{16}$$

for some $\gamma_1 \in [0,1]$, $\gamma_2 > 0$ and $|M_j| < \infty, j \notin J_1$.

The proof is available in our prior work, [23]. In what follows we use the results of the above theorem to analyze the convergence of Eq. (11).

B. Convergence Analysis

We start this section with the following lemma, which will be used in the proof of the main theorem.

Lemma 1. *Under Assumptions* **A0-A3** *and no noise, the product of system matrices,* $\mathbf{P}_k$*'s, in the LTV system, Eq.* (11)*, converges to zero if the condition in Theorem 1 holds.*

Proof. We need to show that Assumptions **B0-B1** can be inferred from **A0-A3**. First, note that by assuming $\beta_1 = \beta$, Assumption **B0** can be immediately verified from Eq. (10). Also, if Assumption **A3** holds, i.e., $\Theta_i \cap \kappa \neq \emptyset$, by using the fact that barycentric coordinates sum to one due to the convexity we can write

$$\sum_{j\in\Theta_i(k)\cap\Omega} p_{ij}(k) = 1 - \sum_{m\in\Theta_i(k)\cap\kappa} b_{im}(k). \tag{17}$$

The right hand side of Eq. (17) is maximized when there is only one anchor among the neighbors of the updating node, and the minimum weight is assigned to this anchor, Eq. (12). This provides an upper bound on the i-th row sum of $\mathbf{P}_k$:

$$\sum_{j\in\Theta_i(k)\cap\Omega} p_{ij}(k) \leq 1 - \alpha < 1, \tag{18}$$

which in turn satisfies Assumption **B1** by choosing $\beta_2 = 1-\alpha$. Subsequently, each slice is completed after all rows of $\mathbf{P}_k$ become strictly sub-stochastic, i.e., all nodes receive anchor information either directly or indirectly, and the asymptotic convergence of Eq. (11) (to zero) follows under the condition, Eq. (16), in Theorem 1. □

We now provide our main result in the following theorem.

Theorem 2. *Consider a network of N mobile agents moving randomly in a bounded region of interest in presence of $M \geq 1$ anchor(s). Under Assumptions* **A0-A3***, Eq.* (11) *asymptotically converges to the true agent locations if the condition in Theorem 1 holds.*

Proof. The following updates give the true agent locations:

$$\mathbf{x}^*_{k+1} = \mathbf{P}_k\mathbf{x}^*_k + \mathbf{B}_k\mathbf{u}_k + \widetilde{\mathbf{x}}_{k+1}. \tag{19}$$

By subtracting Eq. (11) from Eq. (19) we can find the error dynamics as follows

$$\mathbf{e}_{k+1} \triangleq \mathbf{x}^*_{k+1} - \mathbf{x}_{k+1} = \mathbf{P}_k(\mathbf{x}^*_k - \mathbf{x}_k) = \mathbf{P}_k\mathbf{e}_k, \tag{20}$$

which converges to zero if $\lim_{k\to\infty}\prod_{l=0}^{k}\mathbf{P}_l = \mathbf{0}_{N\times N}$, which is in turn followed from Lemma 1 under Assumptions **A0-A3**, and completes the proof. □

V. Effects of noise

As we will show in Section VI, imperfect measurements degrade the performance of the localization algorithm. In this section, we provide modifications to the proposed algorithm to deal with the undesirable effects of noise on motion, $\widehat{\mathbf{x}}^i_k$, and on the distance measurements, $\widehat{d}^{ij}_k$.

A. Minimum contribution

If an agent is located close to the boundaries of a convex hull, noise on distance measurements may lead to false inclusion test results. To address this issue, we make the following assumption:

M1: For any agent, $j \in \Omega$, involved in an update, we assume

$$0 < \alpha' \leq (\mathbf{P}_k)_{i,j}, \qquad \forall k, i \in \Omega, j \in \Theta_i(k)\cap\Omega, \tag{21}$$

where i is the updating agent's index, and α' is the minimum contribution of a neighboring agent. Assumption **M1** states that, agent i does not perform an update unless it is located in a proper position inside a convex hull.

B. Inclusion test error

Noise on the motion may as well lead to inaccurate inclusion test results and imperfect location updates *at each iteration*. To tackle this problem, we propose the following modification to the algorithm:

M2: If the inclusion test is passed at time k by a triangulation set, $\Theta_i(k)$, agent i performs an update only if

$$\epsilon^i_k = \frac{\sum_{j\in\Theta_i(k)} A_{\Theta_i(k)\cup\{i\}\setminus j} - A_{\Theta_i(k)}}{A_{\Theta_i(k)}} < \epsilon, \tag{22}$$

where ϵ^i_k is the *relative inclusion test error* at time k for agent i, and ϵ is a design parameter.

C. Convexity

Finally, we make the following assumption to guarantee the convexity in the updates in the presence of noise:

M3: Suppose the inclusion test is passed at time k by a triangulation set, $\Theta_i(k)$, and Eq. (22) holds. Agent i (randomly) picks one of the neighbors in the triangulation set $\Theta_i(k)$, say agent j, and finds the corresponding weight, a^{ij}_k, as follows:

$$a_{ij}(k) = 1 - \sum_{n\in\Theta_i(k)} a_{in}(k) \tag{23}$$

assuming that $\sum_{n\in\Theta_i(k)} a_{in}(k) < 1$.

VI. Simulations

In this section, we provide the simulation results to illustrate the localization algorithm introduced in Section III. We first consider a network of 5 mobile agents, and 1 anchor. We emphasize that our algorithm works if the anchors are mobile, but in this simulation we consider one anchor in a fixed location. In the beginning, all agents are randomly deployed within a restricted region, which is a 20×20 square. We set the communication radius to $r = 2$. All agents are initially assigned with random location estimates. If an agent finds at least 3 neighbors, it performs the convex hull inclusion test. If the test fails the agent does not perform any update, otherwise it updates its location estimate according to Eq. (8). We set $\alpha_k = 0.2$ to ensure that the agents do not completely forget the past information, and $\alpha = 0.1$ to guarantee a minimum contribution from the anchor when it is involved in an update. Fig. 2 (Left) shows the motion model, i.e., the random trajectories of $N = 5$ mobile agents for the first 20

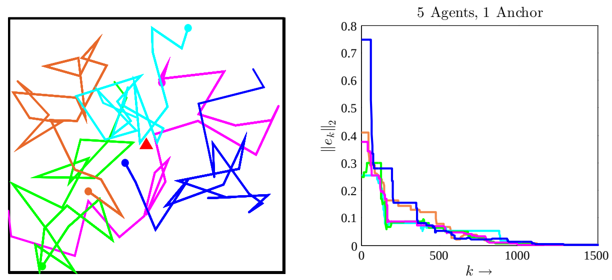

Fig. 2. (Left) Motion model; (Right) Convergence; Red triangle indicate an anchor, and the filled circles show the initial positions of the agents.

iterations. To characterize the convergence, we choose the second norm of the error vector, $\mathbf{e}_k$. The algorithm converges to the true agent locations as $\|\mathbf{e}_k\|_2 \rightarrow 0$. It can be seen in Fig. 2 (Right) that the localization algorithm, tracks the true agent locations.

To illustrate the effect of noise on our localization algorithm, we consider the noise on the motion of agent $i \in \Omega$ at time k, n_k^i, to be $\pm 1\%$ of the magnitude of the motion vector, $\widetilde{\mathbf{x}}_k^i$, and the noise on distance measurements, r_k^{ij} to be $\pm 10\%$ of the actual distance between agent i and node j, d_k^{ij}. As shown in Fig. 3 (Left) this amount of noise leads to an unbounded error on location estimates. However, it can be seen in Fig. 3 (Right) that by applying the modifications, **M1-M3** (with $\epsilon = 20\%$ and $\alpha' = \alpha = 0.1$), to the algorithm, the normalized localization error is reduced to less than 5% for 25 Monte Carlo simulations.

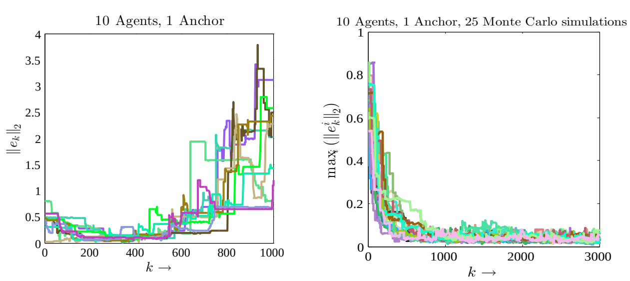

Fig. 3. Effect of noise on the convergence of a network of 10 mobile agents and 1 fixed anchor with $\pm\%10$ noise on distance measurements and $\pm\%1$ noise on the motion; (Left) Original algorithm; (Right) Modified algorithm.

VII. Conclusions

In this paper, we provide a distributed algorithm to track a network of mobile agents moving in a bounded region of interest. Assuming that each agent knows a noisy version of its motion and its distance to the nodes in its communication radius, we provide a *linear* algorithm, which requires at least one anchor to locate an arbitrary number of mobile agents in $\mathbb{R}^m$. The agents update their location estimates as a convex combination of the their $m+1$ neighbors if they lie inside their convex hull. We abstract the algorithm as an LTV system, and show that it converges to the true agent locations under some mild regularity conditions on update weights. We study the impact of noise on the algorithm and provide modifications to counter the undesirable effects of noise.

References

[1] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *Ninth International Conference on Network Protocols*. IEEE, 2001, pp. 35–41.

[2] C. Savarese, J. M. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 4. IEEE, 2001, pp. 2037–2040.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[4] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Information Processing in Sensor Networks*. Springer, 2003, pp. 333–348.

[5] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 166–179.

[6] B. D. Anderson, P. N. Belhumeur, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, and Y. R. Yang, "Graphical properties of easily localizable sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 177–191, 2009.

[7] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based localization of large-scale sensor networks with complex shape," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 4, p. 31, 2009.

[8] U. A. Khan, S. Kar, and J. M. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, May 2009.

[9] Y. Diao, Z. Lin, M. Fu, and H. Zhang, "A new distributed localization method for sensor networks," in *9th Asian Control Conference (ASCC)*, June 2013, pp. 1–6.

[10] Y. Diao, Z. Lin, and M. Fu, "A barycentric coordinate based distributed localization algorithm for sensor networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4760–4771, Sept 2014.

[11] U. A. Khan, S. Kar, and J. M. F. Moura, "DILAND: An algorithm for distributed sensor localization with noisy distance measurements," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1940–1947, Mar. 2010.

[12] U. A. Khan, S. Kar, B. Sinopoli, and J. M. Moura, "Distributed sensor localization in euclidean spaces: Dynamic environments," in *46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 361–366.

[13] J. J. Leonard, H. F. Durrant-Whyte, and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *The International Journal of Robotics Research*, vol. 11, no. 4, pp. 286–298, 1992.

[14] I. J. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 193–204, 1991.

[15] A. Baggio and K. Langendoen, "Monte carlo localization for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 718–733, 2008.

[16] B. Dil, S. Dulman, and P. Havinga, "Range-based localization in mobile sensor networks," in *Wireless Sensor Networks*. Springer, 2006, pp. 164–179.

[17] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 45–57.

[18] J. Yi, S. Yang, and H. Cha, "Multi-hop-based monte carlo localization for mobile sensor networks," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2007, pp. 162–171.

[19] N. Patwari, "Location Estimation in Sensor Networks," Ph.D. thesis, University of Michigan–Ann Arbor, 2005.

[20] Y. G. Kim, J. An, and K. D. Lee, "Localization of mobile robot based on fusion of artificial landmark and RF TDOA distance under indoor sensor network," *International Journal of Advanced Robotic Systems*, vol. 8, no. 4, pp. 203–211, September 2011.

[21] M. J. Sippl and H. A. Scheraga, "Cayley-menger coordinates," *Proceedings of the National Academy of Sciences*, vol. 83, no. 8, pp. 2283–2287, 1986.

[22] A. F. Möbius, *Der barycentrische calcul*, 1827.

[23] S. Safavi and U. A. Khan, "Asymptotic stability of stochastic LTV systems with applications to distributed dynamic fusion," *IEEE Transactions on Automatic Control*, provisionally accepted. [Online]. Available: http://arxiv.org/abs/1412.8018