Localization of Sensor Networks Using Sweeps

Fang, J.; Cao, M.; Morse, A.S.; Anderson, B.D.O.

Published in:
Proceedings of the 45th IEEE Conference on Decision and Control (CDC)

Publication date:
2006

Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

Localization of Sensor Networks Using Sweeps †*

J. Fang
Dept. of Electrical Engineering
Yale University
jia.fang@yale.edu

M. Cao
Dept. of Electrical Engineering
Yale University
m.cao@yale.edu

A. S. Morse
Dept. of Electrical Engineering
Yale University
as.morse@yale.edu

B.D.O. Anderson
Australian National University and National ICT Australia Ltd.
Brian.Anderson@nicta.com.au

Abstract— **The sensor network localization problem with distance information is to determine the positions of all sensors in a network given the positions of some sensors and the distances between some pairs of sensors. We present a specialized localization algorithm and identify the graph properties of some classes of networks that can be localized by the algorithm. We also give an important application of our algorithm in creating formations in multi-agent systems.**

I. Introduction

Determining the positions of sensors is essential in many network applications such as geographic routing, coverage and creating formations. Equipping each sensor in a network with GPS is not feasible in many cases because of the large number of sensors and the cost associated with a GPS unit. Hence, we attack this problem by exploiting the connectivity of a sensor network and some common capabilities of sensors. More specifically, we assume a sensor can measure its distances to and communicate with certain other sensors in the network.

The sensor network localization problem with distance information is to determine the positions of all sensors in a network given the positions of some sensors and the distances between some pairs of sensors. A sensor whose position is given is called an *anchor*. The sensor network localization problem is solvable if and only if the network is "localizable." A network in $\mathbb{R}^d$ is said to be *localizable* if there exists exactly one position in $\mathbb{R}^d$ corresponding to each non-anchor sensor such that the given inter-sensor distances are satisfied. The authors of [1] use rigidity theory to give the necessary and sufficient conditions for a network to be localizable. However, the process of localizing a network has been shown to be NP-hard even when the network is known to be localizable [2]. This leaves us with the more refined questions of how should we go about localizing networks, and what kinds of networks can we efficiently localize? This has been investigated in [3] and we extend the results of that paper. While some ingenious heuristics-based schemes have been proposed ([4], [5], [6], [7]), we are interested in provably correct localization algorithms and the kinds of networks that can be efficiently localized by them. In the following we will give a localization algorithm that consists of a finite number of steps to be carried out sequentially.

We first give some terms and definitions from graph rigidity theory in section 2. We present a localization algorithm in section 3, and identify some classes of networks that can be localized by the algorithm in section 4. We discuss the computational efficiency of the algorithm in section 5, and then give an application of the algorithm in creating formations in section 6.

II. Terms and Definitions

We begin by giving some terms and definitions to be used in the exposition which follows. A *multi-point* $p = \{p_1, \ldots, p_n\}$ in d-dimensional space is a set of n points in $\mathbb{R}^d$ labelled $p_1, \ldots, p_n$. Two multi-points $p = \{p_1, \ldots, p_n\}$ and $q = \{q_1, \ldots, q_n\}$ of n points are *congruent* if for all $i, j \in \{1, \ldots, n\}$, the distance between p_i and p_j is equal to the distance between q_i and q_j. A *point formation* of n points at a multi-point $p = \{p_1, \ldots, p_n\}$ consists of p and a simple undirected graph $\mathbb{G}$ with vertex set $\mathcal{V} = \{1, \ldots, n\}$, and is denoted by $(\mathbb{G}, p)$. If (i, j) is an edge in $\mathbb{G}$, then we say the *length* of edge (i, j) in the point formation $(\mathbb{G}, p)$ is the distance between p_i and p_j.

A point formation $(\mathbb{G}, p)$ is *globally rigid* in $\mathbb{R}^d$ if p and q are congruent multi-points in $\mathbb{R}^d$ whenever $(\mathbb{G}, p)$ and $(\mathbb{G}, q)$ have the same edge lengths. A point formation $(\mathbb{G}, p)$ of n points is *generically globally rigid* in $\mathbb{R}^d$ if there exists $\epsilon > 0$ such that $(\mathbb{G}, q)$ is globally rigid in $\mathbb{R}^d$ for all multi-points $q = \{q_1, \ldots, q_n\}$ where $|q_i - p_i| < \epsilon \ \forall i \in \{1, \ldots, n\}$. A graph $\mathbb{G}$ is said to be *generically globally rigid* in $\mathbb{R}^2$ if $(\mathbb{G}, p)$ is generically globally rigid for some multi-point p. There are a number of efficient algorithms for determining if a graph is generically globally rigid in $\mathbb{R}^2$. If graph $\mathbb{G}$ is

† A preliminary version of this paper has been presented at the 2006 MTNS. A full-length version of this paper with proofs will appear at a later date.

∗ The work of Fang, Cao and Morse was supported in part, by grants from the U.S. Army Research Office and the U.S. National Science Foundation and by a gift from the Xerox Corporation. The work of Anderson was supported by the Research School of Information Sciences and Engineering and National ICT Australia which is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through the Backing Australia's Ability initiative and the ICT Center of Excellence Program.

1-4244-0171-2/06/$20.00 ©2006 IEEE.

generically globally rigid in $\mathbb{R}^2$, then $(\mathbb{G}, p)$ is guaranteed to be globally rigid for all multi-points p in $\mathbb{R}^2$ where the coordinates of p are algebraically independent over the rationals. A graph that is generically globally rigid in $\mathbb{R}^2$ is said to be *minimally generically globally rigid* in $\mathbb{R}^2$ if the removal of any edge causes the graph to not be generically globally rigid in $\mathbb{R}^2$.

A sensor network with n sensors is modelled by a point formation $(\mathbb{G}, p)$, where each sensor corresponds to exactly one vertex of $\mathbb{G}$, and vice versa, with (i, j) being an edge of $\mathbb{G}$ if i and j are distinct and the distance between the corresponding sensors is known, and $p = \{p_1, \ldots, p_n\}$ where p_i is the position of the sensor corresponding to vertex i. We say that $\mathbb{G}$ is the graph of the network, and p is the multi-point of the network. Vertex v of $\mathbb{G}$ is called an *anchor vertex* if the sensor corresponding to v is an anchor, and a *sensor vertex* otherwise. It is known that if the coordinates of the multi-point of a network in $\mathbb{R}^2$ are algebraically independent over the rationals, then the network is localizable if and only if it has at least 3 non-collinear anchors and the graph of the network is generically globally rigid in $\mathbb{R}^2$.

Let $\mathbb{G}$ be a graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$, which we denote by $\mathbb{G} = (\mathcal{V}, \mathcal{E})$. The *second power* of $\mathbb{G}$, written $\mathbb{G}^2$, is the graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E} \cup \mathcal{E}^2$, where $(i, j) \in \mathcal{E}^2$ just in case $i, j \in \mathcal{V}$ and there exists $k \in \mathcal{V}$ such that $(i, k), (k, j) \in \mathcal{E}$. Define a *ring graph* as a graph whose vertices can be labelled as $\{1, \ldots, n\}$ so that vertex i, $1 < i < n$, is adjacent to only vertices $i - 1$ and $i + 1$, vertex 1 is adjacent to only vertices 2 and n, and vertex n is adjacent to only vertices $n - 1$ and 1.

III. The Sweeps Algorithm

In the following, we will describe a localization algorithm in $\mathbb{R}^2$ that consists of a finite number of steps to be carried out sequentially. We will subsequently give examples where the number of steps is at most 2, and separately, at most n.

Let $\mathbb{N}$ be a network of n sensors labelled 1 through n where sensor i is positioned at $\pi(i)$, and $\pi(1), \pi(2), \ldots, \pi(n)$ are distinct points in $\mathbb{R}^2$. Suppose that $1, 2, \ldots, m$ are the labels of $\mathbb{N}$'s anchors and that $m \geq 3$. Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ be the graph of $\mathbb{N}$. Without loss of generality, suppose that for each $i \in \{1, 2, \ldots, n\}$, vertex i of $\mathbb{G}$ corresponds to sensor i and vice versa. For each $v \in \mathcal{V}$, let $\mathcal{N}(v)$ denote the set consisting of all vertices u where $(u, v) \in \mathcal{E}$, and for each $u \in \mathcal{N}(v)$ write d_{uv} for the distance between sensors u and v. By an *assignment* for $\mathbb{N}$ is meant any function $\alpha : \{1, 2, \ldots, n\} \to \mathbb{R}^2$. An assignment for $\mathbb{N}$ is *consistent* if for all $v \in \{1, 2, \ldots, n\}$, $||\alpha(u) - \alpha(v)|| = d_{uv}$ for all $u \in \mathcal{N}(v)$, and $\alpha(v) = \pi(v)$ whenever sensor v is an anchor, i.e. $v \in \{1, 2, \ldots, m\}$. Hence, $\mathbb{N}$ is localizable if and only if there is exactly one consistent assignment for $\mathbb{N}$.

Let $2^{\mathbb{R}^2}$ be the power set of $\mathbb{R}^2$ and write $\mathbb{R}_+$ for the set of positive real numbers. Let $f : 2^{\mathbb{R}^2} \times \mathbb{R}_+ \to 2^{\mathbb{R}^2}$ denote the function $(\mathcal{S}, d) \longmapsto \mathcal{S}'$ where $\mathcal{S}'$ is the set of $p \in \mathbb{R}^2$ such that $||p - q|| = d$ for some $q \in \mathcal{S}$. If $\mathcal{S}$ is not empty, then geometrically $f(\mathcal{S}, d)$ is the union of all points in the plane which lie on circles with the same radius d centered at the points in $\mathcal{S}$. Of course if $\mathcal{S}$ is empty then so is $f(\mathcal{S}, d)$ and conversely. We will be especially interested in the case when $\mathcal{S}$ is a non-empty "finite set" and $d > 0$, where by *finite set* we mean a set with a finite number of points in $\mathbb{R}^2$. In this case $f(\mathcal{S}, d)$ is simply the union of a finite number of circles in the plane which all have radius d. An easily verified property of f is that if $u \in \mathcal{N}(v)$, and $\mathcal{S}(u)$ is a set for which $\pi(u) \in \mathcal{S}(u)$, then $\pi(v) \in f(\mathcal{S}(u), d_{uv})$. We call this the *position keeping* property of f.

Let $\mathbb{S}$ denote the set of all non-empty subsets of $\mathbb{R}^2$ with finitely many elements. Let q be a positive integer no smaller than 2 and write $\mathbb{S}^q$ for the q-fold Cartesian product of $\mathbb{S}$ with itself. Similarly, let $(\mathbb{R}_+)^q$ denote the q-fold Cartesian product of $\mathbb{R}_+$ with itself. Our aim is to define a function $g_q : \mathbb{S}^q \times (\mathbb{R}_+)^q \to 2^{\mathbb{R}^2}$ in such a way so that for each $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q\} \in \mathbb{S}^q$ and $\{d_1, d_2, \ldots, d_q\} \in (\mathbb{R}_+)^q$, $g_q(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q, d_1, d_2, \ldots, d_q)$ is at most a finite set. Furthermore, we shall require the definition of g_q to be such that whenever there are distinct points $u_i \in \mathcal{S}_i$, $i \in \{1, 2, \ldots, q\}$, if $v \in \mathbb{R}^2$ satisfies $\|v - u_i\| = d_i$, $i \in \{1, 2, \ldots, q\}$, then v must be a point in $g_q(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q, d_1, d_2, \ldots, d_q)$. Defining $g_q(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q, d_1, d_2, \ldots, d_q)$ in the most obvious way as the intersection of the sets $f(\mathcal{S}_i, d_i)$, $i \in \{1, 2, \ldots, q\}$, will not be adequate for it may be the case that the resulting intersection is a continuous circle of points in the plane rather than a finite set.

Let $\mathcal{I} = \bigcap_{j=1}^{q} \mathcal{S}_j$, and let $p_1, p_2, \ldots, p_k$ denote the elements of $\mathcal{I}$. For any set $\mathcal{S} \in \mathbb{S}$, and any subset $\mathcal{T} \subset \mathcal{S}$, let $\mathcal{S} \backslash \mathcal{T}$ denote the complement of $\mathcal{T}$ in $\mathcal{S}$. The following definition of g_q satisfies both of the requirements listed above:

$$
\begin{aligned}
&g_q(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q, d_1, d_2, \ldots, d_q) = \\
&\Big(f(\mathcal{S}_1 \backslash \mathcal{I}, d_1) \cap f(\mathcal{S}_2, d_2) \cap \ldots \cap f(\mathcal{S}_q, d_q)\Big) \bigcup \\
&\Big(\bigcup_{i=1}^{k} f(\{p_i\}, d_1) \cap f(\mathcal{S}_2 \backslash \{p_i\}, d_2) \cap \ldots \\
&\qquad \ldots \cap f(\mathcal{S}_q \backslash \{p_i\}, d_q)\Big) \qquad (1)
\end{aligned}
$$

An ordering $v_1, v_2, \ldots, v_n$ of the vertices in $\mathcal{V}$ for which

$$v_i = i,\ i \in \{1, 2, \ldots, m\}$$

and at least one of the sets

$$
\begin{aligned}
\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, v_2, \ldots, v_{i-1}\}, \\
i \in \{m + 1, m + 2, \ldots, n\}
\end{aligned}
$$

is non-empty is called a *sweep* of $\mathbb{N}$ and is denoted by $[v]$. Hence, if an ordering $v_1, v_2, \ldots, v_n$ is a sweep, then there is at least one vertex v_i where $i > m$ and v_i is adjacent to at least one vertex preceding it. We shall require the networks we consider to have at least one "finite position generating" sweep $v_1, v_2, \ldots, v_n$, where by *finite position generating* we mean that each vertex v_i, $i > m$, is adjacent to at least two distinct vertices preceding it. If $v_1, v_2, \ldots, v_n$ is a finite

position generating sweep, then each of the $\mathcal{M}(v_i)$, $i > m$, is a set of at least two elements.

In the following, we will describe a recursive procedure for computing a sequence of finite sets for each vertex $v \in \mathcal{V}$, i.e. $\mathcal{S}(v,1),\ \mathcal{S}(v,2),\cdots\mathcal{S}(v,i),\cdots$, such that each $\mathcal{S}(v,i)$ is a finite set, $\pi(v) \in \mathcal{S}(v,i)$, and if $i < j$, then $\mathcal{S}(v,i) \supset \mathcal{S}(v,j)$. We begin by choosing a finite position generating sweep $[v]$ of $\mathbb{N}$. For $i \in \{m+1, m+2, \ldots, n\}$, let

$$\mathcal{M}(v_i) = \mathcal{N}(v_i) \cap \{v_1, v_2, \ldots, v_{i-1}\}$$

We denote the cardinality of $\mathcal{M}(v_i)$ by q_i and the elements of $\mathcal{M}(v_i)$ by $u_{i1}, u_{i2}, \ldots, u_{iq_i}$. We define the sets $\mathcal{S}(v_i,1),\ i \in \{1,2,\ldots,n\}$ as follows. For $i \in \{1,2,\ldots,m\}$, we define

$$\mathcal{S}(v_i,1) = \{\pi(v_i)\} \tag{2}$$

and for $i \in \{m+1, m+2, \ldots, n\}$, we define

$$\mathcal{S}(v_i,1) = g_{q_i}(\mathcal{S}(u_{i1},1), \mathcal{S}(u_{i2},1), \ldots, \mathcal{S}(u_{iq_i},1), d_{u_{i1}v_i}, d_{u_{i2}v_i}, \ldots, d_{u_{iq_i}v_i}) \tag{3}$$

Note that since $[v]$ is assumed to be a finite position generating sweep, each $\mathcal{M}(v_i)$ has at least 2 elements and so $q_i \geq 2$. Hence, for $i \in \{m+1, m+2, \ldots, n\}$, g_{q_i} is defined and $\mathcal{S}(v_i,1)$ is a finite set because the image of g_{q_i} consists only of finite sets. Since $\mathcal{S}(v_i,1)$, $i \in \{1,2,\ldots,m\}$, are also finite sets because of (2), we have that $\mathcal{S}(v,1)$ is a finite set for each $v \in \mathcal{V}$. Note also that $\pi(v_i) \in \mathcal{S}(v_i,1),\ v_i \in \mathcal{V}$. This is clearly true for $i \in \{1,2,\ldots,m\}$ because of (2). The assumption that $\pi(v)$, $v \in \mathcal{V}$, are distinct together with the definition of g_{q_i} and the position keeping property of f imply that $\pi(v_i) \in \mathcal{S}(v_i,1)$ for $i \in \{m+1, m+2, \ldots, n\}$. We call the computation of $\mathcal{S}(v,1),\ v \in \mathcal{V}$, the *computation of the initial sweep* of $\mathbb{N}$.

Now suppose that the initial sweep of $\mathbb{N}$ has been computed. The sets $\mathcal{S}(v,2),\ v \in \mathcal{V}$, are computed as follows. Let $[u] = u_1, u_2, \ldots, u_n$ be a sweep, and let $\mathcal{M}(u_i) = \mathcal{N}(u_i) \cap \{u_1, u_2, \ldots, u_{i-1}\}$ for $i \in \{m+1, m+2, \ldots, n\}$. Note that $[u]$ need not be a finite position generating sweep. For $i \in \{1,2,\ldots,m\}$ we define

$$\mathcal{S}(u_i,2) = \{\pi(u_i)\} \tag{4}$$

and for $i \in \{m+1, m+2, \ldots, n\}$ we define

$$\mathcal{S}(u_i,2) = \mathcal{S}(u_i,1) \bigcap_{w \in \mathcal{M}(u_i)} f(\mathcal{S}(w,2), d_{wu_i}) \ if\ \mathcal{M}(u_i) \neq \emptyset \tag{5}$$

$$\mathcal{S}(u_i,2) = \mathcal{S}(u_i,1)\ if\ \mathcal{M}(u_i) = \emptyset \tag{6}$$

For each u_i, $i \in \{m+1, m+2, \ldots, n\}$, (5) and (6) implies that $\mathcal{S}(u_i,2)$ must be a finite set since $\mathcal{S}(u_i,1)$ is a finite set. Moreover, since $\mathcal{S}(u_i,2)$, $i \in \{1,2,\ldots,m\}$, are all finite sets because of (4), we have that $\mathcal{S}(v,2)$ is a finite set for each $v \in \mathcal{V}$. Also, $\pi(v) \in \mathcal{S}(v,2)$ for all $v \in \mathcal{V}$. This is clearly true for u_i, $i \in \{1,2,\ldots,m\}$ because of (4). For $i \in \{m+1, m+2, \ldots, n\}$, that $\pi(u_i) \in \mathcal{S}(u_i,1)$ and the position keeping property of f imply $\pi(u_i) \in \mathcal{S}(u_i,2)$. We call the computation of $\mathcal{S}(v,2),\ v \in \mathcal{V}$, the computation of the second sweep of $\mathbb{N}$. It is obvious from (4), (5) and (6) that $\mathcal{S}(v,2) \subset \mathcal{S}(v,1)$ for all $v \in \mathcal{V}$.

Now suppose the kth sweep of $\mathbb{N}$ has been computed, and that for each $v \in \mathcal{V}$, $\pi(v) \in \mathcal{S}(v,k)$ and $\mathcal{S}(v,k)$ is a finite set. The $(k+1)$th sweep is computed as follows. Let $[x] = x_1, x_2, \ldots, x_n$ be a sweep, and let $\mathcal{M}(x_i) = \mathcal{N}(x_i) \cap \{x_1, x_2, \ldots, x_{i-1}\}$ for $i \in \{m+1, m+2, \ldots, n\}$. For $i \in \{1,2,\ldots,m\}$, we define

$$\mathcal{S}(x_i,k+1) = \{\pi(x_i)\} \tag{7}$$

and for $i \in \{m+1, m+2, \ldots, n\}$ we define

$$\mathcal{S}(x_i,k+1) = \mathcal{S}(x_i,k) \bigcap_{w \in \mathcal{M}(x_i)} f(\mathcal{S}(w,k+1), d_{wx_i}) \ if\ \mathcal{M}(x_i) \neq \emptyset \tag{8}$$

$$\mathcal{S}(x_i,k+1) = \mathcal{S}(x_i,k)\ if\ \mathcal{M}(x_i) = \emptyset \tag{9}$$

For each $v \in \mathcal{V}$, we have that $\mathcal{S}(v,k+1)$ is a finite set, $\pi(v) \in \mathcal{S}(v,k+1)$ and $\mathcal{S}(v,k+1) \subset \mathcal{S}(v,k)$ by the same reasoning as before.

The preceding shows that if we compute a sequence of sweeps starting with one which is finite position generating, we can generate a sequence of finite sets for each $v \in \mathcal{V}$, i.e. $\mathcal{S}(v,1)$, $\mathcal{S}(v,2)$, $\cdots$ $\mathcal{S}(v,i)$ $\cdots$, where each set is obtained by means of a finite number of computations and

$$\mathcal{S}(v,1) \supset \mathcal{S}(v,2) \supset \cdots \supset \mathcal{S}(v,i) \cdots$$

and $\pi(v) \in \mathcal{S}(v,i)$ for each i. Thus if we can select a finite number of sweeps, say k, such that for all $v \in \mathcal{V}$, each $\mathcal{S}(v,k)$ will contain just the position $\pi(v)$ of sensor v, then localization will be complete. We call this the *sequential localization* of the network, and we say that the network is *sequentially localizable* in k sweeps. Hence, sequential localization of a network is carried out in a finite number of steps, each of which is solvable in a straightforward manner. This is in sharp contrast to a direct assault on the localization problem by attempting to solve a large number of simultaneous quadratic equations in $2(n-m)$ variables. In the exposition which follows, we will give the graph properties of some networks that are sequentially localizable in just one or two sweeps. The sweeps are selected by considering properties of the network's graph, hence localizing the network in as few sweeps as possible.

In [8], an extension to the Sweeps algorithm is proposed that results in significant reductions in the computational complexity of using Sweeps to localize certain networks. More specifically, suppose $[x]$ is the kth sweep. For x_i, $i \leq m$, let $\mathcal{D}(x_i,k) = \{x_i\}$. For x_i, $i > m$, define $\mathcal{D}(x_i,0) = \{x_i\}$ and let $\mathcal{D}(x_i,k) = \bigcup_{u \in \mathcal{M}(x_i)} \mathcal{D}(u,k) \cup \mathcal{D}(x_i,k-1)$. For x_i, $i \leq m$, define $\mathcal{S}(x_i,k)$ as in (7). For x_i, $i > m$, first let $\mathcal{S}(x_i,k)$ be as defined in (3) if $k = 1$, or (8) and (9) if $k > 1$. For $p \in \mathcal{S}(x_i,k)$, we say that p is a *consistent position* if there is a consistent assignment $\bar{\alpha}$ for the subnetwork of $\mathbb{N}$ corresponding to the subgraph of $\mathbb{G}$ induced by vertices in $\mathcal{D}(x_i,k)$, where $\bar{\alpha}(x_i) = p$ and $\bar{\alpha}(x_j) \in \mathcal{S}(x_j,k)$

for $x_j \in \mathcal{D}(x_i, k)$, $j < i$, and $\bar{\alpha}(x_j) \in \mathcal{S}(x_j, k-1)$ for $x_j \in \mathcal{D}(x_i, k)$, $j > i$. In the extension to Sweeps, all points in $\mathcal{S}(x_i, k)$ that are not consistent positions are removed from $\mathcal{S}(x_i, k)$ before computing any $\mathcal{S}(x_j, k)$, $j > i$.

IV. Graphical Properties of Sequentially Localizable Networks

A graph is a *bilateration graph with bilateration ordering* $v_1, \ldots, v_n$ if its vertices can be relabelled as $v_1, \ldots, v_n$ so that the subgraph induced by v_1 and v_2 is complete, and each v_i, $2 < i$, is adjacent to at least two distinct vertices v_j which "precede" it in the ordering, where by *precede* we mean $j < i$. A necessary condition for a network to be sequentially localizable is that its graph is a bilateration graph. A graph is *k-connected* if the graph remains connected after removing any $k-1$ vertices and the edges incident on those vertices. It is easy to see that bilateration graphs are 2-connected.

All the networks referred to below are in $\mathbb{R}^2$ and the coordinates of their multi-points are assumed to be algebraically independent over the rationals. We note that for any positive integer d, the coordinates of *almost all* multi-points in $\mathbb{R}^d$ are algebraically independent over the rationals.

A. Networks Sequentially Localizable in One and Two Sweeps

A graph is said to be a *trilateration graph with trilateration ordering* $v_1, \ldots, v_n$ if its vertices can be relabelled as $v_1, \ldots, v_n$ so that the subgraph induced by v_1, v_2, v_3 is complete and each v_i with $i > 3$ is adjacent to at least three distinct vertices v_j which precede it in the ordering ([1]). It is shown in [1] that trilateration graphs are generically globally rigid in $\mathbb{R}^2$.

A network is said to be *easily localizable* if its sensors can be relabelled as $v_1, \ldots, v_n$ so that the distance between v_i and v_j is known when $i, j \leq 3$, and the position of sensor v_i, $i > 3$, can be uniquely determined from just the positions of sensors v_j where $j < i$ and the distance between v_j and v_i is known, together with the distances from those sensors to v_i ([1], [3]). It is easy to see that a network with at least three anchors is easily localizable if and only if its graph is a trilateration graph.

Suppose $\mathbb{N}$ has at least three anchors and the graph of $\mathbb{N}$ is a trilateration graph with trilateration ordering $v_1, \ldots, v_n$. If v_1, v_2 and v_3 are anchor vertices, then $\mathbb{N}$ is sequentially localizable in one sweep. Otherwise, $\mathbb{N}$ can be localized using one sweep of the Sweeps algorithm in combination with a simple Euclidean transformation. For complete details, please see the full length version of the paper.

Let $\mathbb{G}$ be the graph of a network. We define the *maximal anchor-free subgraph* of $\mathbb{G}$ to be the maximal subgraph of $\mathbb{G}$ containing only sensor vertices. We say that $\mathbb{G}$ is *partially acyclic* if its maximal anchor-free subgraph is acyclic.

Lemma 1: Let $\mathbb{G}$ be the graph of a network. If $\mathbb{G}$ is generically globally rigid in $\mathbb{R}^2$ and partially acyclic, then $\mathbb{G}$ is a bilateration graph.

Remark 1: A stronger version of lemma 1 also holds. If each sensor vertex of $\mathbb{G}$ has degree at least three and $\mathbb{G}$ is partially acyclic, then $\mathbb{G}$ is a bilateration graph. This is stronger since each vertex of a graph that is generically globally rigid in $\mathbb{R}^2$ must have degree at least three.

Theorem 1: A network with at least three anchors is sequentially localizable in two sweeps if its graph is generically globally rigid in $\mathbb{R}^2$ and partially acyclic.

Remark 2: The network's graph must be generically globally rigid in $\mathbb{R}^2$ in order for theorem 1 to hold. This is because the network cannot be localizable if its graph is not generically globally rigid in $\mathbb{R}^2$, and it is straightforward to show that a sequentially localizable network must also be localizable.

A number of partially acyclic graphs that are generically globally rigid in $\mathbb{R}^2$ are also minimally generically globally rigid in $\mathbb{R}^2$. This and theorem 1 implies there are localizable networks with just enough edges to ensure localizability that are also sequentially localizable.

Many practical networks are such that the distance between two sensors is known if the sensors are within sensing radius of each other. Suppose $\bar{\mathbb{N}}$ is such a network, and let $\bar{\mathbb{G}}$ be the graph of $\bar{\mathbb{N}}$.

Theorem 2: Suppose there exists a subgraph $\mathbb{G}_r$ of $\bar{\mathbb{G}}$ with the same vertex set as $\bar{\mathbb{G}}$, and $\mathbb{G}_r$ is a ring graph with at least three anchor vertices i, j and k where j is adjacent to both i and k. Then $\bar{\mathbb{N}}$ is sequentially localizable in two sweeps after doubling the sensing radius of each sensor.

Remark 3: Theorem 2 can be used to show the following. Suppose there exists a subgraph $\mathbb{G}_r$ of $\bar{\mathbb{G}}$ with the same vertex set as $\bar{\mathbb{G}}$, and $\mathbb{G}_r$ is a ring graph. If $\bar{\mathbb{N}}$ has at least three anchors, then after doubling the sensing radius of each sensor, $\bar{\mathbb{N}}$ is either sequentially localizable in two sweeps or $\bar{\mathbb{N}}$ can be localized using two sweeps of the Sweeps algorithm in combination with a simple Euclidean transformation. For the complete details, see the full length version of the paper.

A graph is *edge 2-connected* if there exists two paths with no edge in common between any two vertices. It is known that the second power of an edge 2-connected graph is generically globally rigid in $\mathbb{R}^2$ ([3]). An important consequence of this and theorem 2 is that if the graph of a network is edge 2-connected with at least three anchor vertices, and the network is such that the distance between two sensors is known if the sensors are within sensing radius, then the network is sequentially localizable after doubling the sensing radius of all the sensors.

B. Networks Sequentially Localizable in at most n Sweeps

Let $\mathbb{N}$ be a network of n sensors labelled 1 through n where sensor i is positioned at $\pi(i)$, and $\pi(1), \pi(2), \ldots, \pi(n)$ are distinct points in $\mathbb{R}^2$. Let $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ denote the graph of $\mathbb{N}$, and suppose $\mathcal{A} = \{1, \ldots, m\}$ is the set of $\mathbb{N}$'s anchors. Without loss of generality, suppose that for each $i \in \{1, 2, \ldots, n\}$, vertex i of $\mathbb{G}$ corresponds to sensor i and vice versa. For each $v \in \mathcal{V}$, let $\mathcal{N}(v)$ denote the set consisting of all vertices u where $(u, v) \in \mathcal{E}$, and for each $u \in \mathcal{N}(v)$ write d_{uv} for the distance between sensors u and v. A sensor v of $\mathbb{N}$ is said to be *localizable* if for all consistent assignments α for $\mathbb{N}$, we have that $\alpha(v) = \pi(v)$ ([9]). Note that if sensor $v \in \mathcal{A}$, then v is localizable by definition. While all sensors in a localizable network are localizable, it is possible to have localizable sensors in a non-localizable network. Using the notions of localizable sensors and partially acyclic graphs, we will define the graph properties of a class of networks that are sequentially localizable in at most n sweeps, where n is the number of sensors in the network.

We say that $\mathbb{G}$ is *recursively acyclic* if its vertices can be ordered as $v_1, v_2, \ldots, v_n$ so that for each v_k, $k \in \{1, 2, \ldots, n\}$, there is a subgraph $\mathbb{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$ of $\mathbb{G}$ that satisfies the following conditions. Before stating the conditions, we give some definitions. For each $k \in \{1, \ldots, n\}$, define $\mathcal{V}_k^{\mathcal{A}} = \{v_i \in \mathcal{V}_k \mid i < k \text{ or } v_i \in \mathcal{A}\}$. For each $k \in \{1, \ldots, n\}$, let $\mathbb{N}_k$ be the network with sensor set $\mathcal{V}_k$ where the position of sensor $v \in \mathcal{V}_k$ is $\pi(v)$. Also, if $v \in \mathcal{V}_k^{\mathcal{A}}$, then the position of sensor v in $\mathbb{N}_k$ is given, and the distance between sensors u and w in $\mathbb{N}_k$ is given whenever $(u, w) \in \mathcal{E}_k$. The conditions are:

1) Each vertex in $\mathcal{V}_k \backslash \mathcal{V}_k^{\mathcal{A}}$ has degree at least three in $\mathbb{G}_k$.

2) The maximal subgraph of $\mathbb{G}_k$ containing only vertices in $\mathcal{V}_k \backslash \mathcal{V}_k^{\mathcal{A}}$ is acyclic.

3) Sensor v_k is a localizable sensor of the network $\mathbb{N}_k$.

Let $\mathcal{N}_1$ denote the class of networks with recursively acyclic graphs. Let $\mathcal{N}_2$ denote the class of all networks $\mathbb{N}$ where the graph of $\mathbb{N}$ is a trilateration graph and $\mathbb{N}$ is sequentially localizable in one sweep. Let $\mathcal{N}_3$ denote the class of all networks $\mathbb{N}$ where $\mathbb{N}$ has at least three anchors and the graph of $\mathbb{N}$ is generically globally rigid in $\mathbb{R}^2$ and partially acyclic. From theorem 1, we have that networks in $\mathcal{N}_3$ are sequentially localizable in two sweeps. It is also straightforward to show that $\mathcal{N}_2$ and $\mathcal{N}_3$ are not disjoint, $\mathcal{N}_2 \not\subset \mathcal{N}_3$ and $\mathcal{N}_3 \not\subset \mathcal{N}_2$.

Theorem 3: 1) If $\mathbb{N} \in \mathcal{N}_1$, then $\mathbb{N}$ is sequentially localizable in at most n sweeps, where n is the number of sensors in $\mathbb{N}$.
2) $\mathcal{N}_2 \subset \mathcal{N}_1$, and the containment is strict, so $\mathcal{N}_1$ contains $\mathcal{N}_2$ as a proper subset.
3) $\mathcal{N}_3 \subset \mathcal{N}_1$, and the containment is strict, so $\mathcal{N}_1$ contains $\mathcal{N}_3$ as a proper subset.

Hence, $\mathcal{N}_2 \cup \mathcal{N}_3 \subset \mathcal{N}_1$, and the class of networks with recursively acyclic graphs gives us a larger and more comprehensive class of networks that are sequentially localizable.

V. Sequential Localization and Computational Efficiency

Sequential localization of a network is not always computationally efficient. However, in some cases it is possible to efficiently localize a network by localizing sections of the network in sequence. For example, let $\mathbb{N}$ be a network, and suppose there are subnetworks $\mathbb{N}_1, \mathbb{N}_2 \ldots \mathbb{N}_D$ of $\mathbb{N}$ such that each sensor of $\mathbb{N}$ is in at least one of the subnetworks. Suppose $\mathbb{N}_1$ is sequentially localizable, and each $\mathbb{N}_i$, $i \in \{2, \ldots, D\}$, is sequentially localizable when each sensor v of $\mathbb{N}_i$ is considered an anchor of $\mathbb{N}_i$ if v is in a subnetwork $\mathbb{N}_j$ where $j < i$. The network $\mathbb{N}$ can be localized by first sequentially localizing $\mathbb{N}_1$, and then sequentially localizing each $\mathbb{N}_i$, $i \in \{2, \ldots, D\}$, once all $\mathbb{N}_j$ where $j < i$ have been sequentially localized. For each $i \in \{1, \ldots, D\}$, let $Z(\mathbb{N}_i)$ denote the computational complexity of sequentially localizing $\mathbb{N}_i$. Then the complexity of sequentially localizing the entire network is determined by D and the $Z(\mathbb{N}_i)$s. If D and each $Z(\mathbb{N}_i)$ are acceptably bounded, then the complexity of sequentially localizing the entire network is rendered acceptable.

In the full length version of the paper, we identify the relationship between the graph properties of networks and the computational complexity of the sequential localization of networks.

VI. Application of Sweeps

Consider a set of agents in the plane where each agent can communicate with and measure its distance to any other agent within sensing range. A formation is given by specifying certain pairs of agents and the desired distance between each specified pair of agents. The agents are said to be *in* a formation if the distance between each pair of agents specified by the formation is the desired distance. Given a particular formation, the goal is for all the agents to position themselves so that they are in the formation. We assume that if the formation specifies a distance δ between agents i and j, then δ is known to both agents i and j, and agents i and j are initially within sensing range.

When the given formation satisfies certain conditions, the Sweeps algorithm can be used to compute a target position for each agent so that the agents will be in formation when the agents are at their target positions. The target positions are all relative positions in the local coordinate system of a particular agent. The selection of this agent as well as the computation of the Sweeps algorithm can be carried out by the agents. Please see the full length paper for the details.

Currently, we are in the process of devising a control strategy for each agent that will cause all of the agents to move to their target positions. Note that the agents do not have GPS and can only measure their distances to and

communicate with other agents within sensing range. Hence, one of the main challenges will be to ensure that each agent remains within sensing range of certain other agents as the agents are moving.

VII. Conclusion

We have presented the localization algorithm Sweeps and identified several classes of networks that can be successfully localized by the algorithm. The next step in our research is to identify the graph properties of all networks that can be efficiently localized by Sweeps. An important part of our future works will also be to analyze the effects of distance measurement errors on the Sweeps algorithm. In this paper we have studied the sensor network localization problem and the Sweeps algorithm in the absence of distance measurement errors. This will help us greatly in pinpointing key issues when we consider the localization problem with distance measurement errors.

References

[1] T. Eren, D. Goldenberg, W. Whiteley, Y.R. Yang, A.S. Morse, B.D.O. Anderson, and P.N. Belhumeur. Rigidity, computation, and randomization in network localization. In *Proc. of IEEE INFOCOM*, 2004.

[2] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. O. Anderson, and P. N. Belhumeur. A theory of network localization. In *Transactions on Mobile Computing*, 2005.

[3] B.D.O. 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. Submitted to Wireless Networks.

[4] A. Savvides, H. Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *ACM Mobile networks and applications*, 8:443–451, 2003.

[5] Joe Albowicz, Alvin Chen, and Lixia Zhang. Recursive position estimation in sensor networks. In *Proceedings of the 9th International Conference on Network Protocols*, pages 35–41, 2001.

[6] L. Doherty, K.S.J. Pister, and L.E. Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings of IEEE INFOCOM*, Anchorage, Ak, 2001.

[7] D. Niculescu and B. Nath. Ad-hoc positioning system. In *Proceedings of IEEE Globecom*, 2001.

[8] D.K. Goldenberg, Pascal Bihler, M. Cao, J. Fang, B.D.O. Anderson, A.S. Morse, and Y.R. Yang. Localization in sparse networks using sweeps. To appear in Mobicom, 2006.

[9] D.K. Goldenberg, A. Krishnamurthy, W.C. Maness, Y.R. Yang, A. Young, A.S. Morse, A. Savvides, and B.D.O. Anderson. Network localization in partially localizable networks. In *Proceedings of IEEE Infocom*, 2005.