Yale University
Department of Computer Science

Rigidity, Computation, and Randomization in Network Localization

Tolga Eren, David K. Goldenberg[1], Walter Whiteley[2],
Y. Richard Yang[3], A. Stephen Morse,
Brian D.O. Anderson[4], Peter N. Belhumeur

YALEU/DCS/TR-1257
July, 2003

[1]This author supported in part by NSF Graduate Research Fellowship DGE0202738.
[2]Supported by grants from NSERC (Canada) and NIH (USA).
[3]This work supported in part by NSF grant ANI-0207399
[4]Supported by the Australian Government through the Department of Communications, Information Technologies and the Arts and by the Australian Research Coucil via a Discovery-project Grant and the Centre of Excellence program.

Rigidity, Computation, and Randomization in Network Localization

T. Eren* D. Goldenberg* W. Whiteley† Y. R. Yang*
A. S. Morse* B. D. O. Anderson‡ P. N. Belhumeur§

*Yale University, New Haven, CT
†York University, Toronto, Ontario
‡Australian National University and National ICT Australia, Canberra
§Columbia University, New York, NY

***Abstract*— In this paper we provide a theoretical foundation for the problem of network localization in which some nodes know their locations and other nodes determine their locations by measuring the distances to their neighbors. We construct grounded graphs to model network localization and apply graph rigidity theory to test the conditions for unique localizability and to construct uniquely localizable networks. We further study the computational complexity of network localization and investigate a subclass of grounded graphs where localization can be computed efficiently. We conclude with a discussion of localization in sensor networks where the sensors are placed randomly.**

I. Introduction

Location service is a basic service of many emerging computing/networking paradigms. For example, in pervasive computing [1], [2], knowing the locations of the computers and the printers in a building will allow a computer to send a printing job to the nearest printer. In sensor networks, the sensor nodes need to know their locations in order to detect and record events, or to route packets using geometric-aware routing, *e.g.*, [3].

One method to determine the location of a node is manual configuration. However, this is unlikely to be feasible for any large-scale deployment or when nodes move often. Another possibility is GPS [4]. However, GPS is costly both in terms of hardware and power requirements. Furthermore, since GPS requires line-of-sight between the receiver and the satellites, it may not work well in buildings or in the presence of obstructions such as dense vegetation, foliage, or mountains blocking the direct view to the GPS satellites.

Recently, novel schemes have been proposed to determine the locations of the nodes in a network where only some special nodes (called beacons) know their locations, *e.g.*, [5], [6]. In these schemes, network nodes measure the distances to their neighbors and then try to determine their locations. The process of computing the locations of the nodes is called *network localization*. For example, in [5], Savvides et al. propose the iterative multilateration scheme to determine the locations of nodes that do not know their locations initially.

Although the designs of the previous schemes have demonstrated great engineering ingenuity, and their effectiveness is verified through extensive simulations, some fundamental questions have not been addressed; as a result, the previous schemes are mainly heuristic-based and a full theoretical foundation of network localization is still lacking.

Specifically, we identify the following two fundamental questions:

1) *What are the conditions for unique network localizability?* Although the network localization problem has already been studied extensively, the precise conditions under which the network localization problem is solvable are not known.
2) *What is the computational complexity of network localization?* Even though the computational complexity of general graph embeddability problem has been studied before [7], the computational complexity of determining the locations of the nodes in a uniquely localizable network has not been studied. Furthermore, constraints on the communication and computational resources of the network nodes may be severe in scenarios such as sensor networks. The localization problem in such resource-constrained settings needs to be investigated.

The objective of this paper is to provide some systematic answers to these two questions. We address the first question using graph rigidity theory and the second question for general uniquely localizable networks and random geometric graphs.

More specifically, in order to answer the first question, we propose *grounded graphs*. In these graphs, each node represents a network node, and two nodes in the graph are connected if the distance between the two is known, *i.e.*, when the distance between the two nodes is measured or when the two nodes are beacon nodes and thus their distance is *implicitly* known. Given our construction of a grounded graph, we show that the network has a unique localization if and only if the corresponding grounded graph is *generically globally rigid*. By observing this connection, we are able to apply the results from the graph-rigidity literature and provide systematic answers to the first question.

We further propose inductive sequences for constructing uniquely localizable networks, both in the plane and in 3-space. By following these sequences, a designer of a network can be assured that the constructed network is uniquely localizable, thus avoiding expensive trial-and-error procedures.

To answer the second question, we study the computational complexity of localization. We prove that even if the grounded graph is a generically globally rigid graph, the problem of determining the locations of the nodes is still NP-hard. To reduce the computational and communication complexity of localization, which may be important in settings such as sensor networks, we study a class of graphs called trilateration graphs. We show that trilateration graphs are uniquely localizable and the locations of the nodes can be computed efficiently. We also show that in a random geometric graph, a network will be a trilateration graph with high probability if a certain node density or communication radius is reached. We provide asymptotic results on the density of the beacons sufficient for trilateration to be carried out in $O(1)$ step, $O(\sqrt{\log(n)})$ steps, or $O(\sqrt{n})$ steps, where n is the number of nodes in the network.

The rest of this paper is organized as follows. The specific network localization problem to be addressed is formulated in Section II. The concepts of rigidity and global rigidity are discussed in Section III. In Section IV, we study the computational complexity of solving the localization problem. In Section V, we study random geometric graphs. In Section VI, we present simulation results. In Section VII, we present related work and compare them with this paper. Our conclusion and future work are in Section VIII.

II. Formulation

A. The Network Localization Problem

In this paper we shall be concerned with the "network localization problem with distance information" which can be formulated as follows. One begins with a network $\mathbf{N}$ in real d-dimensional space (where $d = 2$ or 3) consisting of a set of $m > 0$ nodes labelled 1 through m that represent special "beacon" nodes together with $n - m > 0$ additional nodes labelled $m + 1$ through n that represent ordinary nodes. Each node is located at a fixed position in $\mathbb{R}^d$ and has associated with it a specific set of "neighboring" nodes. Although a node's neighbors are typically defined to be all other nodes within some specified range, other definitions could also be used. The essential property we will require in this paper is that the definition of a neighbor be a symmetric relation on $\{1, 2, \ldots, n\}$ in the sense that node j is a neighbor of node i if and only if node i is also a neighbor of node j. Under these conditions $\mathbf{N}$'s neighbor relationships can be conveniently described by an undirected graph $\mathbb{G}_{\mathbf{N}} = (V, E_{\mathbf{N}})$ with vertex set $V = \{1, 2, \ldots, n\}$ and edge set $E_{\mathbf{N}}$ defined so that (i, j) is one of the graph's edges precisely when nodes i and j are neighbors. We assume throughout that $\mathbb{G}_{\mathbf{N}}$ is a connected graph. The *network localization problem with distance information* is to determine the locations x_i of all nodes in $\mathbb{R}^d$ given the graph of the network $\mathbb{G}_{\mathbf{N}}$, the positions of the beacons $x_j,\ j \in \{1, 2, \ldots, m\}$ in $\mathbb{R}^d$, and the distance $\delta_{\mathbf{N}}(i, j)$ between each neighbor pair $(i, j) \in E_{\mathbf{N}}$.

The network localization problem just formulated is said to be *solvable* if there is exactly one set of vectors $\{x_{m+1}, \ldots x_n\}$ in $\mathbb{R}^d$ consistent with the given data $\mathbb{G}_{\mathbf{N}}$, $\{x_1, x_2, \ldots, x_m\}$, and $\delta_{\mathbf{N}} : E_{\mathbf{N}} \to \mathbb{R}$. In this paper we will be concerned with "generic" solvability of the problem which means, roughly speaking, that the problem should be solvable not only for the given data but also for slightly perturbed but consistent versions of the given data. It is possible to make precise what generic solvability means as follows. Fix $\mathbb{G}_{\mathbf{N}}$ and let $e_1, e_2, \ldots, e_q$ denote the edges in $E_{\mathbf{N}}$. Note that for any set of n points $y_1, y_2, \ldots, y_n$ in $\mathbb{R}^d$ there is a unique distance vector z whose kth component is the distance between y_i and y_j where $(i, j) = e_k$. This means that there is a well-defined function $f : \mathbb{R}^{nd} \to \mathbb{R}^{(md+q)}$ mapping $\{y_1, y_2, \ldots, y_n\} \longmapsto \{y_1, y_2, \ldots, y_m, z\}$. Solvability of the network localization problem is equivalent to f being injective at $\{x_1, x_2, \ldots, x_n\}$ in the sense that the only set of points $\{y_1, y_2, \ldots, y_n\} \in \mathbb{R}^{nd}$ for which $f(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_n)$ is $\{y_1, y_2, \ldots, y_n\} = \{x_1, x_2, \ldots, x_n\}$. In this context it is natural to say that the network localization problem is *generically solvable* at $\{x_1, x_2, \ldots, x_n\}$ if it is solvable at each point in an open neighborhood of $\{x_1, x_2, \ldots, x_n\}$. In other words, the localization problem is solvable at $\{x_1, x_2, \ldots, x_n\}$ if there is an open neighborhood of $\{x_1, x_2, \ldots, x_n\}$ on which f is an injective function.

B. Point Formations

To study the solvability of the network localization problem, we reformulate the problem in terms of a "point formation." As we shall see, the point formation relevant to the network localization problem has associated with it the *grounded graph* of the network, $\hat{\mathbb{G}}_{\mathbf{N}}$, with the same vertices as $\mathbb{G}_{\mathbf{N}}$ but with a slightly larger edge set which adds "links" or edges from every beacon to every other. It is a property of $\hat{\mathbb{G}}_{\mathbf{N}}$ rather than $\mathbb{G}_{\mathbf{N}}$ which proves to be central to solvability of the localization problem under consideration.

We begin by reviewing the point formation concept. By a d-dimensional *point formation* [8] at $p \stackrel{\Delta}{=}$ column $\{p_1, p_2, \ldots, p_n\}$, written $\mathbb{F}_p$, is meant a set of n points $\{p_1, p_2, \ldots, p_n\}$ in $\mathbb{R}^d$ together with a set $\mathcal{L}$ of k *links*, labelled (i, j), where i and j are distinct integers in $\{1, 2, \ldots, n\}$; the *length* of link (i, j) is the Euclidean distance between point p_i and p_j. The idea of a point formation is essentially the same as the concept of a "framework" studied in mathematics [9], [10], [11] as well as within the theory of structures in mechanical and civil engineering. For our purposes, a point formation $\mathbb{F}_p = (\{p_1, p_2, \ldots, p_n\}, \mathcal{L})$ provides a natural high-level model for an n-node network in real 2 or 3 dimensional space. In this context, the points p_i represent the positions of nodes, *i.e.*, both beacons and ordinary nodes, in $\mathbb{R}^d$ and the links in $\mathcal{L}$ label those specific node pairs whose inter-node distances are given. Thus for the network $\mathbf{N}$, $\mathcal{L}$ would consist of all edges in $\hat{\mathbb{G}}_{\mathbf{N}}$, since the distance between every pair of beacons is determined by their specified positions.

$\mathbb{F}_p$ uniquely determines a graph $\mathbb{G}_{\mathbb{F}_p} \stackrel{\Delta}{=} \{V, \mathcal{L}\}$ with vertex set $V \stackrel{\Delta}{=} \{1, 2, \ldots, n\}$ and edge set $\mathcal{L}$, as well as a distance function $\delta : \mathcal{L} \to \mathbb{R}$ whose value at $(i, j) \in \mathcal{L}$ is the distance between p_i and p_j. Let us note that the distance function of $\mathbb{F}_p$ is the same as the distance function of any point formation $\mathbb{F}_q$ with the same graph as $\mathbb{F}_p$ provided q is *congruent* to p in the sense that there is a distance preserving map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T(q_i) = p_i, i \in \{1, 2, \ldots, n\}$. In the next section, we will say that two point formations $\mathbb{F}_p$ and $\mathbb{F}_q$ are *congruent* if they have the same graph and if q and p are congruent. It is clear that $\mathbb{F}_p$ is uniquely determined by its graph and distance function *at most* up to a congruence transformation. A formation that is *exactly* determined up to congruence by its graph and distance function is called "globally rigid." More precisely, a d-dimensional point formation $\mathbb{F}_p$ is said to be *globally rigid* if each d-dimensional point formation $\mathbb{F}_q$ with the same graph and distance function as $\mathbb{F}_p$ is congruent to $\mathbb{F}_p$. It is clear that any formation whose graph is complete is globally rigid. The following simple generalizations of this fact in Lemma 1 provide sufficient conditions for global rigidity that are especially relevant to the network localization problem. In d dimensions, we say a set of points $p_1, \ldots, p_{d+1}$ is *in general position* if it does not lie in a proper subspace (*i.e.*, three points in the plane do not lie on a line, and four points in space do not lie in a plane).

Lemma 1:

1) Let $\mathbb{F}_p = (\{p_1, p_2, \ldots, p_n\}, \mathcal{L})$ be an n-point formation in $\mathbb{R}^2$ that contains three points p_a, p_b, and p_c in general position. Suppose that the graph of the formation $\mathbb{G}_{\mathbb{F}_p}$ contains the complete graph on $\{a, b, c\}$. If the only n-point formation in $\mathbb{R}^2$ that contains these three points and has the same link set as $\mathbb{F}_p$ is $\mathbb{F}_p$ itself, then $\mathbb{F}_p$ is globally rigid.
2) Let $\mathbb{F}_p = (\{p_1, p_2, \ldots, p_n\}, \mathcal{L})$ be an n-point formation in $\mathbb{R}^3$ that contains four points p_a, p_b, p_c, and p_d in general position. Suppose that graph of the formation $\mathbb{G}_{\mathbb{F}_p}$ contains the complete graph on $\{a, b, c, d\}$. If the only n-point formation in $\mathbb{R}^3$ that contains these four points and has the same link set as $\mathbb{F}_p$, is $\mathbb{F}_p$ itself then $\mathbb{F}_p$ is globally rigid.

These properties are direct consequences respectively of the fact that the identity on $\mathbb{R}^2$ is only distance preserving map $T : \mathbb{R}^2 \to \mathbb{R}^2$ that leaves p_a, p_b, and p_c unchanged and the fact that the identity on $\mathbb{R}^3$ is the only distance preserving map $T : \mathbb{R}^3 \to \mathbb{R}^3$ that leaves p_a, p_b, p_c and p_d unchanged. A proof of the lemma will not be given.

C. Solvability of the Network Localization Problem

With the previous definition of point formations, we can now restate the network localization problem in terms of its associated point formation $\mathbb{F}_x$. In the present context, the problem is to determine $\mathbb{F}_x$, given the graph and distance function of $\mathbb{F}_x$ as well as the beacon position vectors $x_1, x_2, \ldots, x_m$. Solvability of the problem demands that $\mathbb{F}_x$ be globally rigid; for if $\mathbb{F}_x$ were not globally rigid it would be impossible to determine $\mathbb{F}_x$ up to congruence, let alone to determine it uniquely. Assuming $\mathbb{F}_x$ is globally rigid, solvability of the network localization problem reduces to making sure that the group of transformations T that leaves the set $\{x_1, x_2, \ldots, x_m\}$ unchanged – namely distance preserving transformations $T : \mathbb{R}^d \to \mathbb{R}^d$ for which $T(x_i) = x_i,\ i \in \{1, 2, \ldots, m\}$ – also leaves unchanged the set $\{x_{m+1}, \ldots, x_n\}$. The easiest way to guarantee this in $\mathbb{R}^2$ is to require $\{x_1, x_2, \ldots, x_m\}$ to contain three points $x_{i_1}, x_{i_2}, x_{i_3}$ in general position; for if this is so, then the only distance preserving transformation that leaves $\{x_1, x_2, \ldots, x_m\}$ unchanged is the identity map on $\mathbb{R}^2$. Similarly, if in $\mathbb{R}^3$, $\{x_1, x_2, \dots, x_m\}$ contains at least four points in general position, then T will again be an identity map, in this case on $\mathbb{R}^3$. We summarize.

Theorem 1: Let $\mathbf{N}$ be a network in $\mathbb{R}^d$, $d = 2$ or 3, consisting of $m > 0$ beacons located at positions $x_1, x_2, \dots, x_m$ and $n - m > 0$ ordinary nodes located at positions $x_{m+1}, \dots, x_n$. Suppose that for the case $d = 2$ there are at least three beacons in general position. Similarly, for the case $d = 3$ suppose there are at least four beacons positioned at points in general position. Let $\mathbb{F}_x$ denote the point formation whose points are at $x_1, x_2, \dots, x_n$ and whose links are those labelled by all neighbor pairs and all beacon pairs in $\mathbf{N}$. Then for both $d = 2$ and $d = 3$ the network localization problem is solvable if and only if $\mathbb{F}_x$ is globally rigid.

III. Rigidity and Global Rigidity

In the previous section, we established that under certain mild conditions, the solvability of the network localization problem is equivalent to the "global rigidity" of point formation. In this section we focus on global rigidity and related concepts from rigidity.

As we have already stated, a d-dimensional point formation $\mathbb{F}_p$ is globally rigid if each d-dimensional point formation $\mathbb{F}_q$ with the same graph and distance function as $\mathbb{F}_p$ is congruent to $\mathbb{F}_p$. In order to clearly present properties of global rigidity, we need several other mathematical concepts whose roots can be found in the rich classical theory of rigid structures.

A. Rigidity

Let $\mathbb{F}_p$ be a d-dimensional point formation, with the distance function measuring all edges in $\mathcal{L}$, $\delta : \mathbb{R}^{nd} \to \mathbb{R}^k$. We are interested in all possible formations with the same distances, that is, in $\delta^{-1}(\delta(p))$. This is a smooth manifold in $\mathbb{R}^{nd}$ [9] and we want to know whether it contains only points congruent to p. Our best tool for studying this manifold is through its tangent space and the matrix equation defining this tangent space with a linearized version of the distance constraints.

For each edge $(i, j) \in \mathcal{L}$, the distance equation $(q_i - q_j)^T(q_i - q_j) = \delta(i, j)^2$ generates the corresponding linear equation

$$(q_i - q_j)^T(\dot{q}_i - \dot{q}_j) = 0$$

in the unknown vector $(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$. If a vector satisfies all these equations, then it lies in the tangent space. This entire system is written as a matrix equation:

$$R(\mathbb{F}_q)\dot{q} = 0, \tag{1}$$

where $\dot{q} = \text{column } (\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$, and $R(\mathbb{F}_q)$ is the specially structured $k \times dn$ array called the *rigidity matrix* of the formation. In structural engineering and mathematics, the solutions $\dot{q}$ are called *first-order flexes* (infinitesimal flexes, or virtual velocities) [9], [10], [11].

The tangent vectors to the congruences of the space $\mathbb{R}^d$ generate a subspace of trivial solutions, called the *trivial flexes*. In the plane, provided that we have at least two distinct points, this space has dimension 3, generated by two translations and the tangent vector to a rotation about the origin. In 3-space, if we have three non-collinear points, this space has dimension 6, generated by three translations along the axes and the derivatives of three rotations about the three axes though the origin. If these trivial flexes are the entire space of first-order flexes, the formation with is *first-order rigid*. In short, provided we have at least three vertices [9], [11]:

Theorem 2: Assume $\mathbb{F}_p$ is a formation with at least d nodes in d-space,

$$\text{rank } R(\mathbb{F}_p) \le \begin{cases} 3n - 6 & \text{if } d = 3 \\ 2n - 3 & \text{if } d = 2. \end{cases}$$

The formation $\mathbb{F}_p$ in the plane is first-order rigid if and only if $\text{rank } R(\mathbb{F}_p) = 2n - 3$. The formation $\mathbb{F}_p$ in 3-space is first-order rigid if and only if $\text{rank } R(\mathbb{F}_p) = 3n - 6$.

It is easy to see from the form of the rigidity matrix that the entries in $R(\mathbb{F}_p)$ are polynomial (actually linear) functions of p. Because of this, the values of p for which the rank of $R(\mathbb{F}_p)$ is below its maximum value form a proper algebraic set in $\mathbb{R}^{dn}$. This observation lies at the roots of the following equivalences [10], [11]:

Theorem 3: Given a formation graph $\mathbb{G}$ with $n \ge 2$ vertices in the plane (resp. $n \ge 3$ vertices in 3-space) the following are equivalent:

1) for some formation $\mathbb{F}_p$ with this graph, $\text{rank } R(\mathbb{F}_p) = 2n - 3$ (resp. $\text{rank } R(\mathbb{F}_p) = 3n - 6$ in 3-space);
2) for all $q \in \mathbb{R}^{2n}$ in an open neighborhood of p, the formation $\mathbb{F}_q$ on the graph $\mathbb{G}$ is first-order rigid in the plane (resp. $q \in \mathbb{R}^{3n}$, $\mathbb{F}_q$ is first-order rigid in 3-space);
3) for all q in an open dense subset of $\mathbb{R}^{2n}$, the formation $\mathbb{F}_q$ on the same graph $\mathbb{G}$ is first-order rigid in the plane (resp. open dense subset of $\mathbb{R}^{3n}$, $\mathbb{F}_q$ is first-order rigid in 3-space);

When property 3) holds, we say that the graph $\mathbb{G}$ of $\mathbb{F}_p$ is *generically rigid* in the space. It is well known that first-order rigidity implies all of the other standard forms of rigidity for a formation, but the converse can fail [9], [10], [12]. For readers thinking of other concepts of rigidity, we point out that if one of these alternative forms of rigidity holds for a non-empty open set, then all of the properties in the Theorem hold [9], [10].

For the plane we have a strong combinatorial characterization of the generically rigid graphs. We note that this leads to a fast ($O(|V|^2)$ algorithm for generic rigidity testing [13].

Theorem 4 (Laman [14]): A graph $\mathbb{G} = (V, \mathcal{L})$ with n vertices is generically rigid in $\mathbb{R}^2$ if and only if $\mathcal{L}$ contains a subset E consisting of $2n - 3$ edges with the property that for any nonempty subset $E' \subset E$, the number of edges in E' cannot exceed $2n' - 3$ where n' is the number of vertices of $\mathbb{G}$ which are endpoints of edges in E'.

There is no comparable complete result for 3-space, and no known polynomial time algorithm, though there are useful partial results [10], [11].

B. Conditions for Global Rigidity

We are interested in the stronger concept of generic global rigidity. This concept is intimately related with first-order rigidity. If the formation $\mathbb{F}_p$ is not first-order rigid, there is a non-trivial first-order flex $\dot{p}$ that does not come from a congruence. This is enough to guarantee that a small perturbation will create a formation that is not globally rigid.

Theorem 5 (Averaging Theorem [10], [11]): Given a non-degenerate formation $\mathbb{F}_p$ with a non-trivial flex $\dot{q}$, the formations $\mathbb{F}_{p+t\dot{q}}$ and $\mathbb{F}_{p-t\dot{q}}$ on the same graph, for all $t > 0$, have the same edge lengths for all links but are not congruent.

We say that a formation $\mathbb{F}_q$ is *generically globally rigid* if every sufficiently small perturbation q of p creates a globally rigid formation $\mathbb{F}_q$. The result above shows that any non-degenerate generically globally rigid formation $\mathbb{F}_p$ must be first-order rigid. However, as Figure 1 illustrates, the converse is not true.

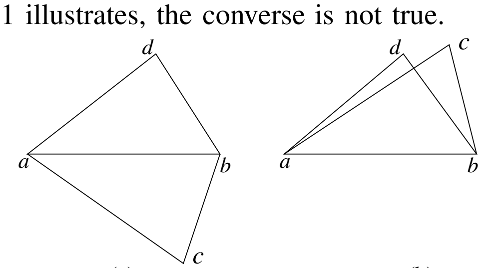

Fig. 1. Two first-order rigid formations with the same graph and the same distance values.

A graph $\mathbb{G}$ is *redundantly rigid* in $\mathbb{R}^d$ if the removal of any single edge results in a graph that is also generically rigid in $\mathbb{R}^d$. As Figure 2 suggests, we need the graph to be generically redundantly rigid to ensure generic global rigidity.

Recall that a graph $\mathbb{G}$ is *k-connected* if it remains connected upon removal of any set of $< k$ vertices. The k-connectivity of a complete graph with n vertices is defined to be $n - 1$. A simple mental check also confirms that for more than $d + 1$ vertices in dimension d, we need at least $d + 1$ vertex connectivity, to avoid a reflection of one component through a mirror placed on a disconnecting set of size d.

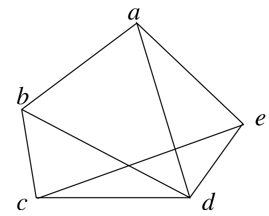

Fig. 2. A globally rigid formation in the plane.

An graph $\mathbb{G} = \{\mathcal{V}, \mathcal{L}\}$ with n vertices is *generically globally rigid* in $\mathbb{R}^d$ if there is an open dense set of points $p \in \mathbb{R}^{dn}$ at which $\mathbb{F}_p$ is a globally rigid formation with link set $\mathcal{L}$. In the plane, a recent result gives a complete characterization of generically globally rigid graphs.

Theorem 6 ([15]): A graph $\mathbb{G}$ with $n \geq 4$ vertices is generically globally rigid in $\mathbb{R}^2$ if and only if it is 3-connected and redundantly rigid in $\mathbb{R}^2$.

Notice that to actually carry out a test to decide whether or not a given graph $\mathbb{G}$ is generically globally rigid in $\mathbb{R}^2$, one must establish that it is both 3-connected and redundantly rigid in $\mathbb{R}^2$. Various tests for 3-connectivity are known, and we refer the reader to [16], [17] for details including measures of the complexity of the tests involved. Tests for redundant rigidity in $\mathbb{R}^2$ have been derived [13] based on variants of Laman's theorem [14].

Since these properties are also required for even a non-empty open set of globally rigid formations in the plane, we can see that the existence of one generically globally rigid formation $\mathbb{F}_p$ implies the graph is generically globally rigid. In 3-space, whether having one generically globally rigid formation is enough to show that the graph is generically globally rigid is an open question [18].

As with generic rigidity, we do not have a generalization of Theorem 6 to higher dimensions. However, it extends as a necessary but not sufficient condition.

Theorem 7 ([13], [18]): If a graph with more than $d + 1$ vertices is generically globally rigid in d-space, then $\mathbb{G}$ is redundantly rigid and at least $d+1$ connected.

In all dimensions $d \geq 3$, there are redundantly rigid and at least $d + 1$ connected graphs that are not generically globally rigid.

It is possible to derive useful sufficient conditions and inductive constructions for generically globally rigid graphs in spaces of all dimensions [12], [18].

One simple construction inserts new nodes of degree $d{+}1$ into existing generically globally rigid formations to create larger generically globally rigid formations. Since we will use this construction later, we give some formal definitions using the term 'trilateration' from the plane as a general term.

Lemma 2: Given a generically globally rigid point formation $\mathbb{F}_p$, and a new point p_0 linked to $d+1$ nodes $p_1, ...p_{d+1}$ of $\mathbb{F}_p$, in general position, then the extended point formation $\bar{\mathbb{F}}_{p+p_0}$ is generically globally rigid.

Proof: Consider any location for the distances in $\bar{\mathbb{F}}_p$. We show that the location of p_0 is unique, given these prior locations.

We first give the proof in $\mathbb{R}^2$, where $\mathbb{F}_p$ has three non-collinear points p_a, p_b, p_c. We have the distances from p_0 to these three points. The distances from the first two points, p_a, p_b, define two intersections of corresponding circles centered at p_a and p_b. The distances from any third point p_c to these two solutions are different, since p_c is not on the line through p_a, p_b. Therefore there is a unique position for p_0 for the given distance to p_c.

The same argument works in all dimensions, starting with the two points of intersection for d spheres with centers in general position.

Now, consider a second formation $\bar{\mathbb{F}}_{q+q_0}$ with the same link lengths. Since the generically globally rigid formation $\mathbb{F}_p$ is contained in this extended formation, the location of its nodes is unique, up to congruence. The unique congruence T defined by the $d+1$ general position points of attachment induces a position $T(p_0)$ that satisfies our construction. Since the constructed point was unique, we conclude that $T(p_0) = q_0$ and the two extended formations are congruent. We conclude that the extended formation is globally rigid.

The general position property used is stable under small perturbations of p. Therefore the global rigidity holds for all small perturbations and the extended formation is generically globally rigid. ■

For the network setting in 2 dimensions, we can start with the globally rigid formation on $m \geq 3$ beacons as $\mathbb{F}_p$. We can then sequentially add new nodes as points $q_{m+1}, \ldots, q_n$, each along with 3 edges to distinct nodes in the existing formation, to make a formation $\bar{\mathbb{F}}_q$. Provided that all sets of points which will be used in extensions are in general position, we create a generically globally rigid formation $\bar{\mathbb{F}}_q$ with n points. This process can be worded in terms of generically globally rigid graphs.

Definition 1: A *trilateration extension* in dimension d of a graph $\mathbb{G} = (V, E)$, where $|V| \geq d+1$ produces a new graph $\mathbb{G}' = (V \cup \{v\}, E \cup \{(v, w_1), \ldots, (v, w_{d+1})\})$, where $v \notin V$, and $w_i \in V$.

Definition 2: A *trilaterative ordering in dimension d* for a graph $\mathbb{G}$ is an ordering of the vertices $1, \ldots, d+1, d+2, \ldots n$ such that $\mathbb{K}_{d+1}$, the complete graph on the initial vertices, is in $\mathbb{G}$, and from every vertex $j > d+1$, there are at least $d+1$ edges to vertices earlier in the sequence. Graphs for which a trilaterative ordering exists in dimension d are called *trilateration graphs in dimension d*.

Theorem 8: Trilateration graphs in d are generically globally rigid in d.

Proof: Any formation on the complete graph on $d+1$ vertices is generically globally rigid if the points are in general position. We take such a formation. We can then apply Lemma 2 to add each point along the trilaterative ordering, with its guaranteed $d+1$ edges, to create a larger generically globally rigid formation with all points in general position. We can then add any additional edges beyond the $d+1$ needed, without changing the generic global rigidity of the extended formation.

Repeated application of this leads to a generically globally rigid formation on the whole graph. Since the conditions of being in general position apply to an open dense subset of the space, we conclude that the graph is generically globally rigid. ■

A trilateration graph $\mathbb{G}$ may have more than one trilaterative ordering and even more than one *seed* - the initial complete graph $\mathbb{K}_{d+1}$. We will look at algorithmic aspects of trilateration graphs in the next section.

IV. Computational Complexity of Localization

We have seen in previous sections that global rigidity is a necessary condition for the solvability of network localization. We will now move from the decision problem of solvability to an associated search problem, graph realization.

Specifically, we define the graph realization problem as the problem of assigning coordinates to vertices of a weighted graph G, so that the edge weight of every edge (i, j) equals the distance between the points assigned to vertices i and j. Note that not all weighted graphs have a realization in the plane. In the context of network localization, the graphs under study are the grounded graphs associated with network point formations.

A. Global Optimization Approaches to Localization

Realization of a graph $\mathbb{G} = (V, E)$ with edge weight function $\delta(i, j)$ can be formulated as a global optimization over vectors of points $\{x_1, x_2, \ldots, x_{|V|}\}$ of the following form,

$$minimize \sum_{(i,j)\in E} (\delta(i,j) - \parallel x_i - x_j \parallel)^2 .$$

This formulation of the problem has been used by biologists studying molecular conformation [19]. Because the optimization is computationally expensive, divide-and-conquer [20] and objective function smoothing [21] strategies have been applied.

B. Computational Complexity of Realizing Globally Rigid Graphs

Although global rigidity testing is computable in polynomial time, and despite the fact that graph realization can be posed as an optimization problem, Saxe has shown that testing the realizability of graphs is NP-hard [7]. Below, we will show that realizing a graph is still NP-hard, even if it is known that the graph is globally rigid and that it has a realization. We first prove a useful NP-hardness result.

Claim 3: The *set-partition-search* problem is the following: Given a set of numbers S with a set-partition, find its set-partition. The set-partition-search problem is NP-hard.

Proof: Assume that algorithm $\mathcal{A}$ solves set-partition-search. Let S be a set of numbers for which it is unknown whether there is a set-partition. Run $\mathcal{A}$ on input S for time t equal to the running time of $\mathcal{A}$ on a valid input of size $|S|$.

If $\mathcal{A}$ has not terminated, then S has no set-partition. If $\mathcal{A}$ has terminated, then S has a set-partition if and only the output of $\mathcal{A}$ is a set-partition of S. Since set-partition is NP-complete, set-partition-search is NP-hard. ■

We now prove another useful result. Note that we will restrict ourselves to two dimensions throughout.

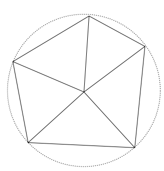

Fig. 3. Wheel graph W_6.

Claim 4: The wheel graph $\mathbb{W}_n$ is globally rigid.

Proof: We will refer to nodes in the cycle, $\mathbb{C}_{n-1}$, as *rim nodes*, the central node as the *hub*, an edge between the hub and a rim node as a *spoke*, and an edge between two rim nodes as a *rim edge*.

If we remove two rim vertices, the graph remains connected through the hub. If we remove the hub and one rim vertex, the graph remains a connected path on the remaining vertices. Therefore removing two vertices does not disconnect the graph, and it is 3-connected.

As Lemma 2.1 of [22] observes, a wheel is a minimally redundantly rigid graph for the plane. By Theorem 6, it is generically globally rigid. ■

We now prove the main result of this section.

Theorem 9: Graph realization for globally rigid weighted graphs that are realizable is NP-hard.

Proof: Assume we have an algorithm $\mathcal{A}$ that takes as input a realizable globally rigid weighted graph and outputs the unique realization. Consider a set of n positive rational numbers $S = \{s_1, s_2, \ldots, s_n\}$, for which a set-partition exists, scaled without loss of generality such that $\sum_{i=1}^{n} s_i \leq \pi/2$. Let us now label the nodes of $\mathbb{W}_{n+1}$ as follows: we label the hub 0, and the rim nodes 1 through n, where there is an edge from i to $i+1$ for $i \in \{1, 2, \ldots, n-1\}$ and from n to 1. We will refer to the spoke from 0 to i as *spoke*$_i$.

Let us now construct a weighted version of $\mathbb{W}_{n+1}$. Let the weight of each spoke be r, where r is a positive rational number. Let the weight of the rim edge between node i and node $i+1$ for $i \in \{1, 2, \ldots, n-1\}$ be $2r\sin(s_i/2)$, and let the weight of the rim edge between node n and node 1 be $2r\sin(s_n/2)$. We now argue that this weighted version of $\mathbb{W}_{n+1}$, call it $\mathbb{W}'_{n+1}$, has a realization in the plane.

If we imagine s_i as the modulus of the angle between spoke$_i$ and spoke$_{i+1}$ for $i \in \{1, 2, \ldots, n-1\}$ and s_n as the modulus of the angle between spoke$_n$ and spoke$_1$ in a realization of $\mathbb{W}_{n+1}$, we can determine a set of edge weights. Fix the weight of each spoke to be r, where r is a positive real number. Then the weight of the rim edge between node i and node $i+1$ for $i \in \{1, 2, \ldots, n-1\}$ must be $2r\sin(s_i/2)$, and the weight of the rim edge between node n and node 1 must be $2r\sin(s_n/2)$. Since S has a set partition, we can form a cycle of these chords in the plane. Therefore the wheel graph with these edge weights, $\mathbb{W}'_{n+1}$ has a realization.

Run algorithm $\mathcal{A}$ on the realizable globally rigid weighted graph $\mathbb{W}'_{n+1}$ obtaining its realization. From this realization, determine the angle s'_i including its sign between spoke$_i$ and spoke$_{i+1}$ for $i \in \{1, \ldots, n-1\}$ and between spoke$_n$ and spoke$_1$. By construction, the set of positive s'_i and the set of negative s'_i are a set-partition of S.

This procedure solves set-partition-search with one call to a graph realization algorithm and polynomial time additional computation. Since set-partition-search is NP-hard, realizable globally rigid weighted graph realization in two dimensions is NP-hard. ■

We will prove a stronger result on the complexity of obtaining an approximate realization of a graph given inaccurate edge lengths in an upcoming revision of this work. This new result is of even more relevance to practical application scenarios, in which the given edge lengths are noisy measurements of the underlying distances.

C. Computational Complexity of Realizing Trilateration Graphs

Although realization of globally rigid graphs is NP-hard, we have already seen a class of globally rigid graphs that are computationally efficient to realize.

Theorem 10: A trilateration graph $\mathbb{G} = (V, E)$ with realizable edge weights is realizable in polynomial time.

Proof: There is a sequence of trilateration extensions that result in $\mathbb{G}$ when applied to $\mathbb{K}_3$. If we know a seed of $\mathbb{G}$, then we can do the following: Localize one of the nodes of the seed at the origin, another on the positive x-axis, and the remaining node at a position with a positive y coordinate. At each step, we can calculate positions for all unlocalized nodes with edges to three localized nodes. Because $\mathbb{G}$ is a trilateration graph, we are guaranteed to be able to calculate positions for all nodes with at most $|V| - 3$ trilaterations.

If we do not know any seed of $\mathbb{G}$, we can guess it in at most $\binom{n}{3}$ tries. A guess is correct if and only if the above procedure succeeds in localizing all nodes in linear time. Hence, we can realize a trilateration graph in polynomial time. ■

As we shall see, there are scenarios in which it is reasonable to assume that we know a seed of the trilateration graph, and in these cases, our linear algorithm will be applicable.

V. Localization in Random Geometric Graphs

In previous sections, we presented theory for localization of general networks. In this section, we specialize to the setting of sensor networks with a large number of randomly distributed sensors. An abstraction that corresponds well to this setting is the random geometric graph.

A. Definition and Properties of Random Geometric Graphs

We define random geometric graphs in terms of point formations.

Definition 5: Given $n \in \mathbb{N}$ and $r \in [0, 1]$, the *random geometric graphs* $\mathbb{G}_n(r)$ are the graphs associated with two dimensional point formations $\mathbb{F}_{|}$ with all links of length less than r, where $p = \{p_1, p_2, \ldots, p_n\}$ is a set of points in $[0, 1]^2$ generated by a two dimensional Poisson point process of intensity n.

The parameters of the model, n and r, correspond respectively to the physical parameters of sensor density and sensing radius.

We next review some useful properties of the connectivity of $\mathbb{G}_n(r)$. Note that the results we present in this section are asymptotic and that because of this, we neglect collinearity as a low probability phenomenon.

As in the case of the Erdös-Rényi random graph model [23], there is a phase transition in the random geometric graph model at which the graph becomes connected with high probability [24]. Penrose [25] generalizes this to k-connectivity with the result that if $\mathbb{G}_n(r)$ has a minimum vertex degree of k then with high probability $\mathbb{G}_n(r)$ is k-connected.

Since it is was proved in [26] for some $r \in O(\sqrt{\frac{\log n}{n}})$, $\mathbb{G}_n(r)$ has a minimum vertex degree of k for $k \in O(1)$ with high probability, $r \in O(\sqrt{\frac{\log n}{n}})$ can also ensure k-connectivity.

B. Global Rigidity of Random Geometric Graphs

Recalling that 3-connectivity is a necessary condition for global rigidity, and using a recent result that 6-connectivity is sufficient for global rigidity in the plane [15], we conclude that $\mathbb{G}_n(r)$ is globally rigid with high probability if and only if $r \in O(\sqrt{\frac{\log n}{n}})$.

An interesting result is that if $\mathbb{G} = (V, E)$ is 2-connected, then the graph $\mathbb{G}^2 = (V, E \cup E^2)$, where E^2 is the set of edges between endpoints of paths consisting of two edges in $\mathbb{G}$, is globally rigid. For random geometric graphs, this means that $\mathbb{G}_n(2r)$ is globally rigid with at least the probability that $\mathbb{G}_n(r)$ is 2-connected.

For some large n and $\delta \in (0, 1)$, let r_i denote the radius at which $\mathbb{G}_n(r)$ is i-connected with probability $1 - \delta$ and let r_g denote the radius at which it is globally rigid with probability $1 - \delta$. Notice that $r_2 \leq r_3 \leq r_g \leq r_6$ and that $r_g \leq 2r_2$.

C. Realization of Random Geometric Graphs

We now explore conditions for $\mathbb{G}_n(r)$ to yield an efficient realization computation.

Theorem 11: If $\lim_{n\to\infty} \frac{nr^2}{\log n} > 8$, with high probability, $\mathbb{G}_n(r)$ is a trilateration graph.

Proof: Partition $[0, 1]^2$ into $\frac{cn}{\log n}$ squares of equal size where $c > 1$. Note that with high probability, every square contains at least three nodes. Additionally, since $r > 2\sqrt{2}\sqrt{\frac{\log n}{n}}$, every node has edges between it and all nodes in its own square and adjacent squares.

Starting from some square we label as 0, we iteratively label every square in $[0, 1]^2$. In step $i \in \{1, \ldots, \sqrt{\frac{cn}{\log n}}\}$, we label with i every unlabelled square that adjoins a square labelled $i - 1$ horizontally, vertically, or diagonally. We will refer to the union of all squares with the same label i as a *layer*, L_i.

We now iteratively label all n nodes in the grid. In step -1, we choose three nodes in L_0 and label them 1, 2, and 3. In step 0, we label the rest of the nodes in L_0 with numbers greater than 3. In step i, we label all nodes in L_i with numbers larger than every label in L_{i-1}.

Every node in L_0 with a label greater than three has edges to 1, 2, and 3. By construction, a node labelled m in L_i, $i > 0$ has edges to at least three nodes in L_{i-1} with labels less than m. Thus we have a trilaterative ordering from Definition 2, and $\mathbb{G}_n(r)$ is a trilateration graph. ■

An intuitive argument that perhaps yields more insight into the previous result is that assuming in the limit of large n that nodes 1, 2, and 3 can be considered to occur at a single point p_0, if every node in $\mathbb{G}_n(r)$ is connected to three other nodes closer than itself to p_0, then $\mathbb{G}_n(r)$ has a trilaterative ordering. Since p_0 can be in any direction from an arbitrary point, this is assured in the event that every node has three neighbors in any $120°$ sector of the circle with radius r about it, or at least nine neighbors. Denoting by r_t the radius at which $\mathbb{G}_n(r)$ has probability $1-\delta$ of being a trilateration graph, we suspect that r_t approaches r_9 from above in the limit of large n.

These results immediately yield insight into the complexity of realizing $\mathbb{G}_n(r)$.

Theorem 12: For some $r \in O(\sqrt{\frac{\log n}{n}})$, if the positions of three nodes with edges to each other are known, then with high probability, a realization of $\mathbb{G}_n(r)$ is computable in linear time.

Proof: By the proof of Theorem 11, the three nodes with known positions form the seed of a spanning trilateration graph $\mathbb{G}$ with high probability. By Theorem 10, the positions of all nodes in $\mathbb{G}$ can be computed in linear time. Since $\mathbb{G}_n(r)$ is spanned by $\mathbb{G}$, it can be realized in linear time. ■

D. Localization in Random Sensor Networks

We now propose a simple localization protocol for random sensor networks called ITP in figure 4. Theorem 12 allows us to analyze the effectiveness of our procedure.

Definition 6: A *random sensornet* $\mathbb{S}_n(r)$ is a sensornet of n sensors with sensing radius r placed at random on $[0,1]^2$ by a two-dimensional Poisson point process. A *beacon* is a sensor that knows its position.

One could define a random sensornet in terms of a uniform distribution over $[0,1]^2$, but we do not consider this case.

The following results are summarized in Table I.

Claim 7: For some $r \in O(\sqrt{\frac{\log n}{n}})$, with high probability, all sensors in $\mathbb{S}_n(r)$ will have determined their positions with ITP by $O(\sqrt{\frac{n}{\log n}})$ time if three beacons are placed anywhere in $[0,1]^2$ so that they are in sensing range of each other.

```
▷ Sensors have two modes: localized and unlocalized
▷ Sensors determine distance from heard transmitter
▷ All sensors are pre-placed and plugged-in

Localized mode:
        Broadcast position
Unlocalized mode:
        Listen for broadcast
        if broadcast from (x,y) heard
           Determine distance to (x,y)
        if three broadcasts heard
           Determine position
           Switch to localized mode
```

Fig. 4. The iterative trilateration protocol (ITP).

Proof: We set r and partition $[0,1]^2$ into labelled squares as in the proof of Theorem 11, this time so that the three beacons are in the square labelled 0. We say that a layer is localized when all sensors in that layer have determined their positions. Assuming ITP broadcast, distance calculation, and trilateration take place in constant time, L_0 will be localized in a single constant-time step. Additionally, given L_i localized, ITP will localize L_{i+1} in a single constant-time step. Therefore, all layers will be localized in at most $O(\sqrt{\frac{n}{\log n}})$ steps and our claim is established. ■

Claim 8: For some $r \in O(\sqrt{\frac{\log n}{n}})$, with high probability, all sensors in $\mathbb{S}_n(r)$ can determine their positions with ITP and will have done so by expected time of at most $O(\sqrt{\log n})$ if beacons are placed on $[0,1]^2$ by a Poisson point process of intensity $O(n/\log n)$.

Proof: We set r and partition $[0,1]^2$ into squares of area A as in the proof of Theorem 11. The Poisson point process places beacons into each square at a rate $\lambda \propto nA/\log n \in O(1)$. Therefore, the probability that a square contains at least three beacons is a constant p.

The probability that all squares contains less than three beacons is $q^{O(n/logn)}$, where $q = 1-p$, so some square contains at least three beacons with high probability, and consequently, all sensors can localize as in claim 7.

We now bound the expected time it takes for every sensor to localize given some square contains three beacons. We say a square is localized if every sensor it contains has determined its position. In a single constant-time step, ITP localizes a square if it contains three beacons or if any of its neighbors are localized. Because of this, in what follows we will refer to discretized time rather than steps.

The probability that a square does not localize by time k is the probability that all squares within a square of squares with side $2k+1$ contain fewer than three beacons, $q^{(2k+1)^2}$. The probability that the last square to localize does so after a certain time is less than the probability that at least one of the squares localizes after that time. More formally, where t_i is the time at which square i localizes, since the number of squares is $O(\frac{n}{\log n})$, the following is true,

$$Pr[max(t_i) > k] \leq min(1, O(\frac{n}{\log n})q^{O(k^2)}).$$

Therefore, the expected time for all squares to localize satisfies the inequality

$$E[max(t_i)] \leq \sum_{i=0}^{\infty} min(1, O(\frac{n}{\log n})q^{O(k^2)}).$$

Observing that for some $k_0 \in O(\sqrt{\log n - \log\log n})$,

$$O(\frac{n}{\log n})q^{O(k^2)} > 1 \iff k < k_0,$$

we see that

$$E[max(t_i)] \leq O(\sqrt{\log n}) + O(\frac{n}{\log n}) \sum_{i=k_0}^{\infty} q^{O(k^2)}.$$

In calculations we will not include here, it can be shown that $O(\frac{n}{\log n}) \sum_{i=k_0}^{\infty} q^{O(k^2)} \in O(1)$.

We have thus shown that with high probability, all sensors will localize in expected time at most $O(\sqrt{\log n})$. ■

Claim 9: For some $r \in O(\sqrt{\frac{\log n}{n}})$, with high probability, all sensors in $\mathbb{S}_n(r)$ can determine their positions and will have determined their positions by $O(1)$ time if beacons are placed on $[0,1]^2$ by a Poisson point process of intensity $O(n)$.

Proof: If $r \in O(\sqrt{\frac{\log n}{n}})$, the Poisson point process places beacons in the sensing region of a sensor at rate $\lambda \propto nr^2 \propto \log n$. Since we expect $O(\log n)$ beacons connected to every sensor, with high probability, we will have $O(1)$ *i.e.*at least three beacons connected to every sensor, and all sensors will localize in $O(1)$ time with high probability. ■

beacons	sensing radius	E[t_{loc}]
$O(1)$	$O(\sqrt{\frac{\log n}{n}})$	$O(\sqrt{\frac{n}{\log n}})$
$O(\frac{n}{\log n})$	$O(\sqrt{\frac{\log n}{n}})$	$O(\sqrt{\log n})$
$O(n)$	$O(\sqrt{\frac{\log n}{n}})$	$O(1)$

TABLE I

LOCALIZATION IN VARIOUS BEACON PLACEMENT SCHEMES.

VI. Evaluations

We simulate random geometric graphs in three dimensions by generating points randomly in $[0,1]^3$, placing four beacons in the center of the unit cube within sensing range of each other. We then simulate ITP by localizing nodes in computational rounds in which we determine positions for all nodes connected to four nodes with known position. We terminate the simulation when a round does not determine the position of any node.

In our first simulation, for three values of r, we track the percentage of nodes whose positions can be determined. Consistent with our prediction of the behavior of ITP using random graph theory, we observe in figure 5 an increasingly sharp phase transition in the percentage of localizable nodes as we increase n.

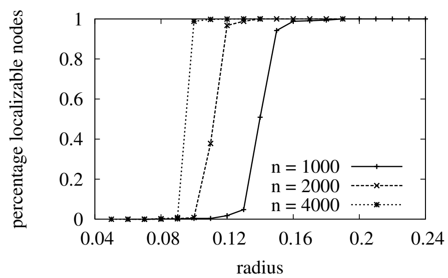

Fig. 5. Percentage of nodes localizable with 4-beacon ITP.

In our second simulation, we calculate the smallest radius at which the percentage of localizable nodes is greater than 95%. We see behavior similar to that predicted by theory in Figure 6. Note that the asymptotic result more accurately models actual behavior as n increases. The difference for small n is explained by the contribution of logarithmic terms in the localization probability that becomes significant when n is small.

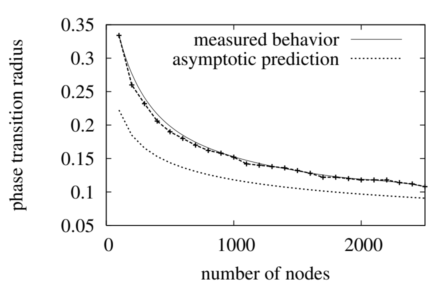

Fig. 6. Trilateration graph phase transition radius in $G_n(r)$.

Our last simulations investigate the number of computational rounds necessary to localize all nodes that can be localized. In figure 7, we observe for $n = 2000$ that the percentage of localized nodes at a given step increases dramatically with modest increases in sensing radius. Note that below the phase transition, at $r = 0.1$, the procedure fails to localize practically any nodes and completes in four steps. For r straddling the phase transition, Figure 8 plots the number of steps before completion. The spike is due to a sudden increase in connectedness above the phase transition at which the radius is minimal for total localizability.

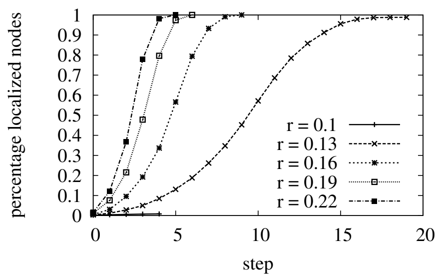

Fig. 7. Time-evolution of the number of localized nodes.

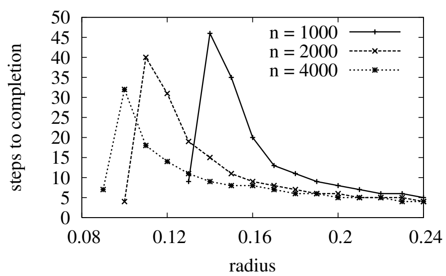

Fig. 8. Required steps for algorithm completion.

VII. Related Work

Network localization is an active research field, *e.g.*, [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [5], [40], [6], [41]. The previous approaches can be classified into two types: coarse-grained and fine-grained. The focus of this paper is fine-grained localization. As we discussed in the Introduction, the previous approaches are mainly heuristics, and this paper provides the first theoretical analysis of network localization.

A related problem called molecular conformation has been studied in the chemistry community, *e.g.*, [20], [21], [42]. However, the focus of these studies is on 3D. Also, since the structure of a molecule is given, they do not consider the network construction process.

One major building block of our analysis is rigidity theory and computational geometry. Rigidity has been long studied in mathematics and structural engineering (see for example [14], [10], [43], [9], [11]) and has a surprising number of applications in many areas.

We formally analyzed the performance of network localization. Even though some researchers have studied random graphs in sensor networks, *e.g.*, [44], [45], [46], [47], their focus is mainly routing and not localization. In [46], Díaz, Petit and Serna analyzed the performance of localization for optical sensor networks. However, their analysis is for the case in which a sensor can derive its position from a single beacon.

VIII. Conclusion and Future Work

The unique localization of networks from distance measurements shares a number of features with work in several other active fields of study: rigidity and global rigidity in frameworks; the coordination formations of automonous agents; and geometric constraints in CAD. In this paper, we have drawn on techniques and results from the first two fields, also combined in some previous joint work [12], as well as specific results on global rigidity [15], [18]. With these concepts, we were able to lay a coherent solid foundation for the underlying problem of when a network is uniquely localizable, for almost all locations of the points. Specifically, we constructed a formation and then a graph for each network such that the localization problem for the network is uniquely solvable, almost always, if and only if the corresponding graph is generically globally rigid. From these connections, we drew specific results and showed that the trilateration networks are uniquely localizable for almost all initial locations.

It should be noted that as stated, the localization problem is not in general numerically well posed since even if it is solvable with the given data, it may be unsolvable with data arbitrarily "close" to that which is given. In practical terms, this means that special attention must be paid to the computation process and to assessing the significance of "approximate solutions." It also means that only graphs which are generically globally rigid are capable of having computationally stable solutions for given data sets. This confirms our choice of conceptual framework for this problem.

We also studied the computational complexity of network localization and presented computational properties of trilateration graphs in that context. For networks based on random locations and communication radius r, we provided some necessary and some sufficient conditions for the network to be uniquely localizable with high probability. We have included some computational evaluations of these techniques.

Other work such as [41] approaches network localization with angles, bearings and headings in addition to some distance constraints. Drawing on more general work on geometric constraints such as angles and directions in CAD, we have further generic global uniqueness results that can give new insights where certain patterns of angles or headings are used [12], as well as insights into the complexity of general patterns of angle constraints. This will be explored further in a future paper.

References

[1] M. Weiser, "Some computer science problems in ubiquitous computing," *Communications of ACM*, July 1993.

[2] G. H. Forman and J. Zahorjan, "The challenges of mobile computing," *IEEE Computer*, vol. 27, no. 4, pp. 38–47, Apr. 1994.

[3] Brad Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of The Sixth International Conference on Mobile Computing and Networking 2000*, Boston, MA, Aug. 2000.

[4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice, Fourth Edition*, Springer-Verlag, 1997.

[5] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of The Seventh International Conference on Mobile Computing and Networking 2001*, Rome, Italy, July 2001, pp. 166–179.

[6] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes in large scale sensor networks," in *Proceedings of The Eighth International Conference on Mobile Computing and Networking 2003*, San Diego, Sept 2003, pp. 81–95.

[7] J.B. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," in *Proceedings of the 17th Allerton Conference in Communications, Control and Computing*, 1979, pp. 480–489.

[8] T. Eren, P.N. Belhumeur, B.D.O. Anderson, and A.S. Morse, "A framework for maintaining formations based on rigidity," *Proceedings of the 15th IFAC World Congress, Barcelona, Spain*, 2002.

[9] B. Roth, "Rigid and flexible frameworks," *American Mathematical Monthly*, vol. 88, pp. 6–21, 1981.

[10] W. Whiteley, "Some matroids from discrete applied geometry," in *Contemporary Mathematics*, James G. Oxley Joseph E. Bonin and Brigitte Servatius, Eds., vol. 197. American Mathematical Society, 1996.

[11] W. Whiteley, "Rigidity and scene analysis," in *Handbook of Discrete and Computational Geometry*, J. Goodman and J. O'Rourke, Eds., pp. 893–916. CRC Press, 1997.

[12] T. Eren, W. Whiteley, A.S. Morse, and P.N. Belhumeur, "Sensor and network topologies of formations with distance-direction-angle constraints," *Submitted to the 42nd IEEE Conference on Decision and Control, Hawaii*, 2002.

[13] B. Hendrickson, "Conditions for unique graph realizations," *SIAM J. Comput.*, vol. 21(1), pp. 65–84, 1992.

[14] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, pp. 331–340, 2002.

[15] Bill Jackson and Tibor Jordàn, "Connected rigidity matroids and unique realizations of graphs," Tech. Rep. TR-2002-12, Eotvos University, Budapest, Hungary, Mar. 2003.

[16] J. E. Hopcroft and R.E. Tarjan, "Dividing a graph into triconnected components," *SIAM J. Comput.*, vol. 3, pp. 135–158, 1973.

[17] G. L. Miller and V. Ramachandran, "A new graph triconnectivity algorithm and its parallelization," *Combinatorica*, vol. 12, pp. 53–76, 1992.

[18] R. Connelly, "On generic global rigidity," *preprint*, 2003.

[19] G.M. Crippen and T.F. Havel, *Distance Geometry and Molecular Conformation*, John Wiley & Sons, 1988.

[20] Bruce Hendrickson, "The molecule problem: Exploiting structure in global optimization," *SIAM Journal on Optimization*, vol. 5, no. 4, pp. 835–857, 1995.

[21] J. More and Z. Wu, "Global continuation for distance geometry problems," *SIAM Journal on Optimization*, vol. 7, pp. 814–836, 1997.

[22] Alex Berg and Tibor Jordan, "A proof of Connelly's conjecture on 3-connected generic cycles," *J. Comb. Theory B.*, 2002.

[23] Bela Bollobás, *Random Graph Theory*, Academic Press, London, 1985.

[24] M.J. Appel and R.P. Russo, "The maximum vertex degree of a graph on uniform points in $[0,1]^2$," *Adv. Applied Probability*, vol. 29, pp. 567–581, 1997.

[25] Mathew D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures and Algorithms*, vol. 15, no. 2, pp. 145–164, 1999.

[26] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Chih-Wei Yi, "Fault tolerant deployment and topology control in wireless networks," in *Proceedings of The ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2003*, Annapolis, MD, June 2003.

[27] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons, "The active badge location system," Tech. Rep. 92.1, Olivetti Research Ltd. (ORL), 24a Trumpington Street, Cambridge CB2 1QA, 1992.

[28] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, 1997.

[29] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71–78, October 1998.

[30] Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of The Fifth International Conference on Mobile Computing and Networking 1999*, Seattle, WA, Nov. 1999, pp. 263–270.

[31] Paramvir Bahl and Venkata N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[32] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications Magazine*, vol. 7, no. 5, pp. 28–34, October 2000.

[33] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The cricket location-support system," in *Proceedings of The Sixth International Conference on Mobile Computing and Networking 2000*, Boston, MA, Aug. 2000, pp. 32–43.

[34] Srdan Capkun, Maher Hamdi, and Jean-Pierre Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *HICSS*, 2001.

[35] Jeffrey Hightower and Gaetano Borriella, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[36] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *IEEE/RSI Int. Conf. on Intelligent Robots and Systems (IROS)*, 2001.

[37] Joe Albowicz, Alvin Chen, and Lixia Zhang, "Recursive position estimation in sensor networks," in *Proceedings of the 9th International Conference on Network Protocols '01*, Riverside, CA, November 2001, pp. 35–41.

[38] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of IEEE INFOCOM '01*, Anchorage, AK, Apr. 2001, pp. 1655–1633.

[39] D. Niculescu and B. Nath., "Ad-hoc positioning system.," in *Proceedings of IEEE Globecom 2001*, November 2001.

[40] C. Savarese, J. Rabay, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *USENIX Technical Annual Conference*, Monterey, CA, June 2002.

[41] Dragos Niculescu and Badri Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, Apr. 2003.

[42] Abdo Y. Alfakih, Amir Khandani, and Henry Wolkowicz, "Solving Euclidean distance matrix completion problems via

semidefinite programming," *Comput. Optim. Appl.*, vol. 12, no. 1-3, pp. 13–30, 1999.

[43] L. Henneberg, *Die Graphische Statik der starren Systeme*, Johnson Reprint, 1911.

[44] Bhaskar Krishnamachari, Stephen B. Wicker, and Ramon Bejar, "Phase transition phenomena in wireless ad-hoc networks," in *Proceedings of Symposium on Ad-Hoc Wireless Networks, GlobeCom 2001*, Antonio, Texas, 2001.

[45] J. Díaz, J. Petit, and M. Serna, "Random scaled sector graphs," Tech. Rep. LSI-02-47-R, Departament de Llenguatges i Sistemes Inform'atics, Universitat Polit'ecnica de Catalunya, June 2002.

[46] J. Díaz, J. Petit, and M. Serna, "A random graph model for optical networks of sensors," Tech. Rep. LSI-02-72-R, Departament de Llenguatges i Sistemes Inform'atics, Universitat Polit'ecnica de Catalunya, December 2002.

[47] Andrs Farag, "Scalable analysis and design of ad hoc networks via random graph theory," in *Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*. 2002, pp. 43–50, ACM Press.