Auto-localization in Local Positioning Systems: a closed-form range-only solution

J. Guevara*, A.R. Jiménez*, A.S. Morse†, J. Fang† C. Prieto*and F. Seco*
*Instituto de Automática Industrial. Consejo superior de Ctra. Campo Real km 0.2, La Poveda,28500, Arganda del Rey.
Madrid (Spain)
Telephone: (34) 918711900, Fax: (34)
†Dept. of Electrical Engineering
Yale University
New Haven, CT 06520 USA

***Abstract*—This paper studies the problem of determining the position of beacon nodes in Local Positioning Systems, for which there is no inter-node distance measurements available. Also, neither the mobile node nor any of the stationary nodes have positioning or odometry information. The common solution is implemented using a mobile node capable of measuring its distance to the stationary beacon nodes within a sensing radius. Many authors had implemented heuristic methods based on optimization algorithms to solve the problem, however such techniques can fail if the range measurements doesn't provide enough information to obtain a unique solution. Equally, the actual methods require a good initial estimation of the node positions in order to find the correct solution. In this paper we use rigidity theory to determine the necessary conditions in which such problem is solvable for Local Positioning Systems. We also a present a new method to calculate the inter-beacon distances based in the linearization of the trilateration equations. This method doesn't require any initial estimation of the nodes position. The simulation results show a good estimation of the beacon nodes position without using any optimization algorithm.**

I. INTRODUCTION

There are many applications that require localization systems in indoor environments, where the Global Positioning System (GPS) is not available. Localization systems designed to work in indoor places are known as Local Positioning Systems (LPS).These systems require the installation of several nodes at fixed positions (called beacon nodes) in the indoor environment. The determination of the beacons position is done manually using measurement tapes to estimate the distance to the closet wall of the building. This procedure of locating the nodes is a cumbersome and error prone procedure and therefore different techniques have been proposed to address the problem of obtaining the position of the nodes, also known as the auto-calibration or auto-localization problem.

Typical solutions are based on the measured ranges from a group of localized nodes to the beacons with unknown position. With enough measurements all the beacons can be localized by trilateration techniques [1]. In [2] four different positions are used to obtain the beacon nodes positions of a 3D LPS system. In [3] three nodes with known positions are required plus a group of nodes with unknown positions. In [4] only the relative distances between four nodes are required. These methods, however, require a external localization system which is not always available in indoors environments. So in order to address a more generic problem it is preferable to assume that no information of the position of any node is known.

Duff y Muller [5] proposed a method to solve the multilateration equations by means of a nonlinear least-square optimization algorithm when no positions are known. The algorithm is based in a degree of freedom analysis, arguing that with enough measurements to the beacon nodes at different positions, enough equations can be obtained to solve all the beacons positions. In [6] the same principle is used replacing the optimization algorithm with an Extended Kalman Filter. However, the degree of freedom analysis does not guarantee that there is a unique solution in a system of nonlinear equations, such as the trilateration equations, when the only data available is the distance measured between the mobile node and the beacons.

In [7] Priyantha et al. introduced the concept of rigidity in LPS systems to develop a method that provides a unique solution of the auto-localization problem. The process requires a starting subset of nodes (beacons and mobile nodes) that were verified to have a unique solution. Then an incremental process of adding more nodes, based in the rigidity theory, is used to preserve the uniqueness of the solution. Priyantha also developed a group of movement strategies to obtain the starting subset of nodes, but these strategies were justified using a degree of freedom analysis. The rigidity theory has been widely used in the sensor networks localization problem [8], although it has not been fully applied on the LPS systems case, probably because the lack of inter-node distance (connectivity) between beacon nodes. Also, until recently rigidity theory was only well developed in 2D problems [9] since the necessary and sufficient conditions for rigidity in 3D were unknown. However, the further development in the rigidity theory allow us to apply this theory to the particular case of LPS systems.

Another problem presented with the optimization algorithm, or Extended Kalman Filter, is that the convergence of both depend heavily on the initial conditions used. A proposed solution to this problem uses multiples initial seeds (estimated position of the nodes) and then selects the best solution based

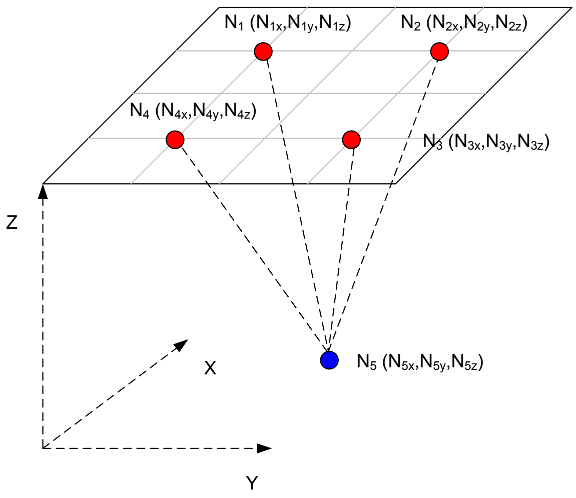

Fig. 1. 3D local positioning system

on a group of objective functions [5]. Other technique employ a heuristic optimization method, based on the simulated annealing algorithm, to estimate the initial position of the nodes [10]. In both cases the algorithms could sometimes fall in a local minima.

In the present paper we use the rigidity theory to verify the solvability of the auto-localization problem in LPS systems. We also propose a new closed-form solution for the initial estimation of the beacon nodes. This paper is organized as follows. Section II describes the auto-localization problem when no position information is known of any node. Section III-A briefly describes the rigidity theory and how it can be used to verify the solvability of the auto-localization problem in localization networks. Section IV presents the precise conditions to be meet in order to solve the auto-localization problem in LPS systems. In section V a new algorithm to obtain the node positions that doesn't require any initial estimations is presented. The proposed solution is then evaluated in section VI.

II. The auto-localization problem

Figure 1 shows a typical LPS system, composed by a group four static nodes, called beacon nodes, and one mobile node. If the beacon positions are known, then the mobile node coordinates (N_{5x}, N_{5y}, N_{5z}) can be calculated using the distance equations, known as trilateration or multilateration equations:

$$\begin{aligned}(N_{1x}-N_{5x})^2+(N_{1y}+N_{5y})^2+(N_{1z}-N_{5z})^2 &= d_{15}^2\\ (N_{2x}-N_{5x})^2+(N_{2y}-N_{5y})^2+(N_{2z}-N_{5z})^2 &= d_{25}^2\\ &\vdots\\ (N_{4x}-N_{5x})^2+(N_{4y}-N_{5y})^2+(N_{4z}-N_{5z})^2 &= d_{45}^2 \quad (1)\end{aligned}$$

Where (N_{ix}, N_{iy}, N_{iz}) $i = 1,..,4$ are the coordinates of the beacon node N_i and d_{i5} the measured distance between the beacon node N_i to the mobile node. If the only data available is the distance measurements, the auto-localization problem can be modeled as a distance optimization problem, where the objective function are the distance equations (1) and the variables are the coordinates of all nodes. A degree of freedom analysis is usually used to verify if the auto-localization problem is solvable. For example, consider a 3D LPS system with f beacon nodes and m mobile points of measurements. The points of measurements, also called virtual nodes, are the coordinates where the mobile node measures its distance to the beacon nodes. The total number of variables for the 3D LPS system is given by the number of coordinates of all nodes minus six coordinates used to define a global coordinate system. The number of distance equations is equal to m times the number of points of measurements, if all beacon are always at range with the mobile node. In order to obtain the position of all nodes the number of distance equations must be at least equal to the number of variables:

$$3(f+m)-6 \leq fm \quad (2)$$

The degree of freedom analysis of (2) is a necessary but not sufficient condition to have a unique solution for all nodes positions. Redundant measurements are normally used to minimize this problem. However, in order to evaluate if a particular node is localizable the precise conditions for a unique solution are required. In this paper we use rigidity theory to verify the solvability of the auto-localization problem. Rigidity theory has been widely used in Ad-hoc networks problems [11] and we be applied it to the LPS case. A main difference between Ad-hoc networks and LPS systems is that range measurements between beacon nodes (or anchor nodes as called in Ad-hoc networks) are not available in the latter. Hence more restrictive conditions are required to obtain a solvable network for LPS systems.

III. Solvability in network localization systems

The rigidity graph theory studies when a given graph, with known number of vertices and edges, has a unique realization. Applying rigidity to the auto-localization problem means that the solvability of it can be verified based only on the total number of beacons and the number of range measurements available between them. In the next sections we will present the basic notions of the theory.

A. Basics of the rigidity theory

In order to apply the rigidity theory to the localization problem we have to represent the LPS system as a undirected graph. Let's represent a LPS system with the graph $G = V, E$ where the n vertices V represent the nodes of the system (beacon nodes and virtual nodes), and the edges E, represent the availability of range measurements between the nodes. Each edge E_{ij} is associated with an edge length d_{ij} which represents the range measurement between nodes N_i and N_j where $i, j = 1..n$. A solution of the positions of the nodes is called a configuration $P(V) = (p_1, \ldots, p_n)$, where $p_i = (N_{ix}, N_{iy}, N_{iz})$ represents the position of the node

N_i. A framework $G(P)$ refers to a graph G along with a configuration P. A configuration is called generic if the coordinates are algebraically independent over the rational, i.e. no three points form line, no four points form a plane in the 3D space, etc. In this paper we will assume that the node locations are generic. Depending of the geometry of the graph G it can have more than one configuration (more than one solution) for a given set of distance measurements d_{ij}.

The network-localizability problem, in rigidity theory, refers to determining if a given graph G with a set of range measurements d_{ij} has only one associated configuration $P(V)$. If that is the case we will say that the network is solvable. Once a network is verified to be solvable the next step is to obtain the respective configuration $P(V)$, i.e. the nodes coordinates. This is called the network localization problem with distance information. In the following sections the basics of the rigidity theory are presented for the 2D and 3D cases.

B. Conditions for unique realization in $\mathbb{R}^2$

A framework $G(P)$ is called **flexible** if exists a continuous deformation from the given configuration P to another P' so that the edge lengths are preserved as shown in figure 2-a. If no such deformation exists, then the graph is called **rigid**. Obviously a flexible graph will generate multiple solutions of the node positions, therefore a graph with unique realization has to be rigid.

The rigidity of a graph depends only on its connectivity and can be verified using the Laman's theorem:

Theorem 3.1: [12] A graph G with n vertices and $2n - 3$ edges is rigid in $\mathbb{R}^2$ if and only if no subgraph has more than $2n' - 3$ edges where n' is the number of vertices of the subgraph.
It is important to note that for $\mathbb{R}^d$ $d \geq 3$ the Laman's theorem is a necessary but not sufficient condition.

Laman's theory states that at least $2n-3$ edges are required to obtain a rigid graph. This can be explained with a degree of freedom analysis: every structure in the plane have three internal degrees of freedom that corresponds to translations and rotations. In addition, every vertex have two degrees of freedom in $\mathbb{R}^2$. Since trivial motions (translation and rotation) does not imply the deformation of the graph (the relative positions between vertices are preserved), the total number of degrees of freedom for flexibility are $2n-3$. Since every edge removes, at most, one degree freedom it is necessary at least $2n-3$ edges to obtain rigidity. However, such condition is no sufficient to obtain a rigid graph, the edges must also be well distributed in order to remove all the degrees of freedom. Such distribution is given by the second condition of the Laman's theorem which verifies the number of edges included in every subgraph of G.

A rigid graph can also have more than one realization in form of partial reflections (figure 2-b). The conditions in which such reflections occurs are given by theorem 3.2:

Theorem 3.2: [12] A rigid graph positioned generically in $\mathbb{R}^d$ will have a partial reflection if and only if it is not vertex (d+1)-connected

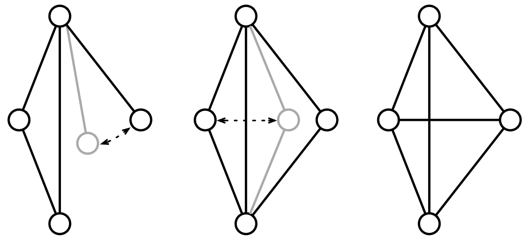

Fig. 2. Examples of different types of graphs based on their rigidity

The $(d+1)$ **connectivity** requirement is well know in the localization problem where in order to obtain a unique position of a mobile node it is necessary the range measurements of at least $(d+1)$ beacon nodes.

Finally theorems 3.1 and 3.2 does not ensure a unique realization of the graph and is necessary to introduce a third requirement called **redundant rigidity** (figure 2-c):

Definition 3.1: [12] A rigid graph positioned generically in $\mathbb{R}^d$ is redundantly rigid if after the removal of any single edge $e \in E$ the remaining graph is also rigid in $\mathbb{R}^d$

Clearly the requirement of redundant rigidity also includes rigidity as defined in theorem 3.1. Finally we can give a definition of the uniqueness of realization a graph, called **global rigidity** (figure 2-c) based in theorem 3.2 and definition 3.1

Theorem 3.3: [12] A graph is generically globally rigid in $\mathbb{R}^2$ if and only if it is 3-connected and redundantly rigid in $\mathbb{R}^2$

Theorem 3.3 shows that the uniqueness in the solution at $\mathbb{R}^2$ is a property of the graph, and only depends on the connectivity between the nodes, not the geometry.

We can test if a graph is globally rigid by verifying if it is 3-connected and redundantly rigid. The first condition is easily verifiable, and the latter condition can be tested by eliminating one by one a single edge of the graph and verifying if the graph is still rigid using Laman's theorem. Several other test algorithms of rigidity had been implemented based in Laman's theorem, a comparison between different algorithms can be found in [13].

Other methods to verify global rigidity in $\mathbb{R}^2$ rely on the inductive construction of the graphs being tested. Beginning with a known globally rigid graph, a sequence of global rigidity preserving operations can be used to obtain the tested graph, therefore showing that such graph is also globally rigid. For example, the (d+1)-connectivity provide a method to construct globally rigid graphs:

Theorem 3.4: [7] A graph is globally rigid in $\mathbb{R}^d$ if it is formed by a starting globally rigid graph at which more (d+1)-connected vertices had been added

Theorem 3.4 is valid for $\mathbb{R}^2$ and $\mathbb{R}^3$. Also other operations that preserves global rigidity can be used such as edge splitting, vertex addition and vertex splitting. A further explanation of these operations can be found in [14]

C. Conditions for unique realization in $\mathbb{R}^d$, $d \geq 3$

From a recent work in [15] it is known that global rigidity of point formations in all dimensions is a property of the graph. This means that the global rigidity of graph in $\mathbb{R}^d$, $d \geq 3$ can be tested, although there is currently no efficiently checkable graphical characterization as in $\mathbb{R}^2$. In order to evaluate if a graph is globally rigid in $\mathbb{R}^d$, $d \geq 3$, first we need to introduce the notion of equilibrium stress [16].

For a given graph $\mathbb{G} = (V, E)$ we can assign to each edge $E_{i,j} \in E$ a weight ω_{ij}, also known as stress. We say that a vertex $i \in V$ is in equilibrium with respect to the stress vector $\omega = (\cdots, \omega_{ij}, \cdots)$ if the values of ω_{ij} are such that:

$$\sum_{j=1}^{n} \omega_{ij}(p_j - p_i) = 0 \tag{3}$$

where $p(i)$ represent a given position of the vertices V_i in $\mathbb{R}^d$.

The stress vector can be arranged into a $n \times n$ symmetric matrix Ω, known as the stress matrix, where the off-diagonal entries are denoted as $-\omega_{ij}$ and the following conditions hold:

1) $\omega_{ij} = 0$ when $i \neq j$ and i, j is not in E
2) $[1, 1, \cdots, 1]\Omega = 0$

The stress matrix can then be used as a sufficient condition to check global rigidity as proposed by Connelly:

Theorem 3.5: [17] If P is a generic configuration in $\mathbb{R}^d$, such that there is a stress, where the rank of the associated stress matrix Ω is $n - (d + 1)$, then the framework $G(P)$ is globally rigid in $\mathbb{R}^d$.

Later Gortler et al [15] showed that this condition is also necessary, implying that the global rigidity is a generic property of the graph. Using these results, Gortler developed a numerical algorithm based on the stress matrix to analyze if a given graph is globally rigid. The algorithm is further explained in the next section.

IV. Solvability of the LPS auto-localization problem

In this section we will apply the rigidity theory to obtain the necessary and sufficient conditions to solve the auto-localization problem in LPS systems. As stated before the LPS system is composed by a group of uniquely identifiable static nodes (the beacon nodes), and a mobile node which is able to measure the distance to the static nodes within a specified radius. No positioning or odometric information is available in the nodes . Given a group of distances measurements between the mobile node to the static nodes, we want to determine the positions of all nodes. Before estimating the node positions the solvability of the problem must be verified. The problem is solvable if with the available distance measurements we can determine the coordinates of the nodes without ambiguity. If the problem is no solvable, there are multiple solutions for the node positions that cannot be detected without adding more information to the problem.

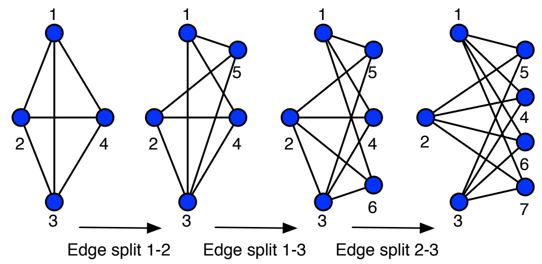

Fig. 3. Sequence of edge splitting operations to demonstrate the rigidity of $K(3,4)$

To apply the rigidity theory we will represent the LPS system using a specific subtype of undirected graphs called bipartite graph. A **bipartite graph** is a graph whose vertices are divided in two groups, where each vertex is connected only to the nodes of the other group. Clearly for LPS systems these groups are divided by beacon nodes and virtual nodes, since the only range measurement available are between the beacons and the mobile node. A bipartite graph is called **complete bipartite graph** when all the vertices of one group are connected to all the vertices of the other. We will use the notation $K(f, m)$ to represent a complete bipartite graph composed by f static nodes N_i $i = 1..f$ and m virtual nodes N_j $j = f + 1..f + m$.

A. Solvability in $\mathbb{R}^2$

In the following, we will assume the stationary nodes and the mobile node are all situated in the plane. We will begin with a system composed by the minimum necessary number of beacon nodes. By a trilateration point of view we know that a necessary condition for $\mathbb{R}^2$ localization is that there are at least three beacon nodes. We need now to determine how many virtual nodes are required in order to obtain a globally rigid graphic. Let's star with a general case of the two dimensional version proposed in [7], which was justified using a degree of freedom analysis, composed by three beacon nodes and four virtual nodes. In other words the complete bipartite graph $K(3, 4)$. Using the inductive construction method with the edge splitting operation we show that such bipartite graph is rigid.

Edge splitting consists in replacing any edge E_{ij} by a new node V_k connected by the edges E_{ki}, E_{kj} and E_{kz} where $z \neq u, v$. The edge splitting operation has been shown to keep the global rigidity in $\mathbb{R}^2$ [17]. Figure 3 shows with a sequence of edge splitting operations on the globally rigid graph $K(4)$ (composed with four vertices all connected) how we can obtain the bipartite graph $K(3, 4)$, therefore demonstrating that the $K(3, 4)$ graph is globally rigid in $\mathbb{R}^2$.

We can now evaluate other bipartite graphs configurations based in the $K(3, 4)$ graph. For example, for a system with four static nodes we have that:

Proposition 4.1: For a system composed by $f = 4$ static nodes, at least three virtual nodes are required to obtain a globally rigid graph.

Because there is no functional distinction, in a graph representation, between static and virtual nodes $K(4,3) \equiv K(3,4)$, and therefore $K(4,3)$ is also a rigid graph.

From theorem 3.4 we know that adding any number of virtual 3-connected nodes to the $K(3,4)$ will conserve the global rigidity of the graph. Therefore we can add as many virtual nodes to the system as long they are 3-connected to the static nodes. The resultant redundancy in measurements can then be used to improve the estimation of the node positions. It is also possible to add more static nodes to generalize the proposition 4.1:

Proposition 4.2: For a system composed by $f \geq 4$ static nodes, at least three virtual nodes are required to obtain a globally rigid graph, i.e $K(f,3)$ for $f \geq 4$.

Proposition 4.2 allows to localize an extended number of beacon nodes with few virtual ones. This, however, is an impractical configuration because requires that all beacon nodes are always in range with the mobile one. Instead an iterative approximation can be used for an extended number of beacon nodes. First we obtain a globally rigid graph with three beacon nodes and then we can add a new beacon node following the next procedure:

Proposition 4.3: To add a static node to a globally rigid graph, at least three virtual nodes are required to maintain global rigidity. The virtual nodes must be 3-connected to the rigid graph.

Finally a similar procedure as proposition 4.3 can be used to connect two globally rigid graphs. This case can be useful when the nodes are distributed in separated areas, e.g. two adjacent rooms.

Proposition 4.4: To connect two globally rigid graphs, at least three virtual nodes are required in order to maintain global rigidity. The virtual nodes must be 3-connected to both rigid graphs.

B. Conditions for solvability in $\mathbb{R}^d$, $d \geq 3$

Although there is currently no efficiently checkable graphical method to verify if a graph is globally rigid in $\mathbb{R}^d$ where $d \geq 3$, we can use a numerical algorithm proposed by Gortler et al [15]. They also demonstrated that the algorithm generates no false positives and very few false negatives. The following algorithm is based on the theorem 3.5 and uses the stress matrix in order to verify global rigidity:

1) Pick a framework P for a graph G with random vertex coordinates
2) Find a stress vector ω solving (3)
3) Create the stress matrix Ω with the previous obtained vector ω
4) If the rank of $\Omega = n - (d+1)$ declare G globally rigid

At least 4 static nodes are required for localization in $\mathbb{R}^3$, so we need to find how many virtual nodes are necessary to form a globally rigid graph. Using the matrix stress test we verified that the bipartite graph K(4,7) is globally rigid.

If we add more beacon nodes the number of necessary virtual nodes are reduced. For example, for 5 static nodes 6 virtual nodes are necessary, and with 6 statics nodes only 5 are required. For 7 or more static nodes we obtain the next condition:

Proposition 4.5: For a system composed by $f \geq 7$ in $\mathbb{R}^d$ $d \geq 3$, at least four virtual nodes are required to obtain a globally rigid graph, i.e $K(f,4)$ for $f \geq 3$.

Finally propositions 4.3 and 4.4 can be extended for 3D:

Proposition 4.6: To add a static node to a globally rigid graph in $\mathbb{R}^d$ $d \geq 3$, at least four virtual nodes are required to maintain global rigidity. The virtual nodes must be (d+1)-connected to the rigid graph.

Proposition 4.7: To connect two globally rigid graphs in $\mathbb{R}^d$ $d \geq 3$, at least four virtual nodes are required to maintain global rigidity. The virtual nodes must be (d+1)-connected to both rigid graphs.

V. Linear solution for the auto-localization problem

Once all the conditions for a unique solution of the auto-localization problem had been verified, the next step is to estimate the beacons positions using the available distance measurements. Due to the nonlinearity of the distance equations an iterative optimization algorithm is usually employed. However these algorithms require a good first estimation of the nodes position in order to avoid being trapped in local minima. By contrast, we propose here a closed-form solution of the auto-localization problem that doesn't require any initial estimate. The algorithm is based in the linearization of the trilateration equations by expanding those equations and grouping all nonlinear terms in additional variables. This approach requires a mayor number of virtual nodes to solve the new added variables. This, however, doesn't represent a real drawback since the optimization methods also use redundant measurements to improve the solutions.

We begin with the simplest LPS configuration for 2D with unique solution, which is composed of three beacon nodes N_i where $i = 1..f$ $(f = 3)$ and at m virtual nodes N_j where $j = m+1..m+f$ $(f \geq 4)$. Without loss of generality, we can choose a coordinate system where the coordinates of N_1 are $(0,0)$ and of N_2 are $(N_2x, 0)$. Next, using the distance equations between beacon nodes, the trilateration equations can be rewritten as a function of two groups of distances measurements: the inter-beacon distances d_{12}, d_{13}, d_{23}, which are the unknown variables, and the distances between beacon nodes and virtual nodes $d_{14}, d_{24}, d_{34}..., d_{3f+m}$, which are the available data. The final equation can be expressed in the linear form $AX = B$ with $f = 3$ beacon nodes and $m = 6$ virtual nodes:

$$X = \begin{bmatrix} d_{13}^2 + d_{23}^2 - d_{12}^2 \\ \frac{d_{13}^2}{d_{12}^2}(d_{12}^2 + d_{23}^2 - d_{13}^2) \\ \frac{d_{23}^2}{d_{12}^2}(d_{12}^2 + d_{13}^2 - d_{23}^2) \\ \frac{d_{13}^2}{d_{12}^2} \\ \frac{d_{23}^2}{d_{12}^2} \\ d_{13}^2 d_{12}^2 \end{bmatrix} \tag{4}$$

$$B = \begin{bmatrix} D_{314} \\ D_{315} \\ \vdots \\ D_{319} \end{bmatrix} \quad (5)$$

$$A = \begin{bmatrix} d_{34}^2 & d_{24}^2 & d_{14}^2 & D_{324}D_{214} & -D_{314}D_{214} & -1 \\ d_{35}^2 & d_{25}^2 & d_{15}^2 & D_{325}D_{215} & -D_{315}D_{215} & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{39}^2 & d_{29}^2 & d_{19}^2 & D_{329}D_{219} & -D_{319}D_{219} & -1 \end{bmatrix} \quad (6)$$

$$D_{ijk} = d_{ik}^2 - d_{jk}^2, \quad i,j = 1,2,3. \quad k = 4,5,\cdots,9. \quad (7)$$

The matrix A and the vector B are known, composed with the distance measurements from the mobile node. The vector X include the unknown inter-node distances. Since the dimension of the vector X is six, at least six virtual nodes are required to solve the linearized equations, but more virtual nodes can be added in order to improve the solution. Once obtained X the unknown variables d_{12}, d_{13}, d_{23} can be calculated by:

$$d_{12} = \sqrt{\frac{\frac{X_2}{X_4} + \frac{X_3}{X_5}}{2}} \quad (8)$$

$$d_{13} = \sqrt{\frac{X_1 + \frac{X_3}{X_5}}{2}} \quad (9)$$

$$d_{23} = \sqrt{\frac{X_1 + \frac{X_2}{X_4}}{2}} \quad (10)$$

Finally, when the inter-node distances are obtained, the beacon nodes coordinates can be calculated using them:

$$\begin{aligned} N_1 &= (0,0) \\ N_2 &= (d_{12},0) \\ N_3 &= \left(\frac{d_{12}^2+d_{13}^2-d_{23}^2}{2d_{12}}, \sqrt{d_{13}^2 - \left(\frac{d_{12}^2+d_{13}^2-d_{23}^2}{2d_{12}}\right)^2} \right) \end{aligned} \quad (11)$$

Because of the complexity added by a third coordinate, the 3D case cannot be solved with the same technique used in 2D. The inter-beacon distances cannot be obtained from the vector X using linear algebra. However, the 2D solution can be used in a 3D LPS system if all the beacon nodes are positioned in a plane. Such configuration is not uncommon since in many LPS systems the beacon nodes are positioned at the ceiling at the same height. Another requirement is that all virtual nodes must be positioned in a plane parallel to the beacons plane. This can be achieved by moving the mobile node at the same height with regard to the floor level. Since the 3D solution requires distribution of the nodes which is not generic, i.e. the nodes are positioned in two different planes, multiple solutions might occur in the form of partial reflections of the virtual nodes at both sides of the beacons plane. This problem be can eliminated by choosing the height coordinate z of the virtual nodes to be always positive.

VI. Simulations and Results

In order to evaluate the performance of the proposed method, a LPS system was simulated using MATLAB. A group of four beacon nodes where placed at the points: (0,0,0), (300,0,0), (300,120,0) and (300,300,0). A group of 21 virtual nodes were placed at a distance of 250 cm from the beacon's plane. They were distributed in a circular pattern in order to avoid that three or more points form a line. All the nodes positions are unknown. The ranging data was generated with a Gaussian noise with zero mean added. The value of the standard deviation of the measurement's noise goes from 0 to 5 cm , and for each noise level 100 simulations with different distance measurements were performed. The performance of the proposed method was evaluated using the average deviation of the beacon localization error:

$$e_p = \frac{1}{n}\sum_{i=1}^{n} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (z_i - \hat{z}_i)^2} \quad (12)$$

where (x_i, y_i, z_i) are the real coordinates of beacon i and $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ are the estimated ones.

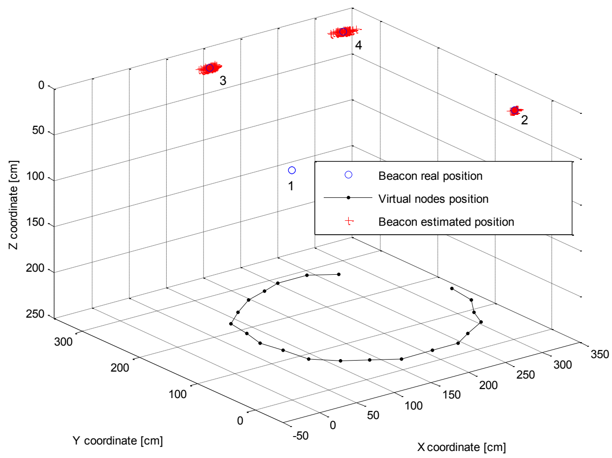

Fig. 4. Simulated localization scheme with a noise level of 2 cm

Figure 4 shows the scheme of the simulated LPS system. Since our algorithm only calculates the position of 3 beacon nodes, the beacons were divided in two subgroups of 3 nodes: nodes $1, 2, 3$ and nodes $1, 2, 4$. The two subgroups positions where calculated independently and later referenced to the same global coordinate system. Since we can choose any subgroup of three nodes, we can have redundant inter-beacon distance estimations that can be used to improve further the estimations. Figure 4 also compares the calculated positions of the beacon nodes with their real positions. A 2 cm noise standard deviation was used. Since node N_1 is always used as the origin of coordinates it doesn't have any positioning error.

The simulation shows that all the other beacon nodes were localizated with good accuracy.

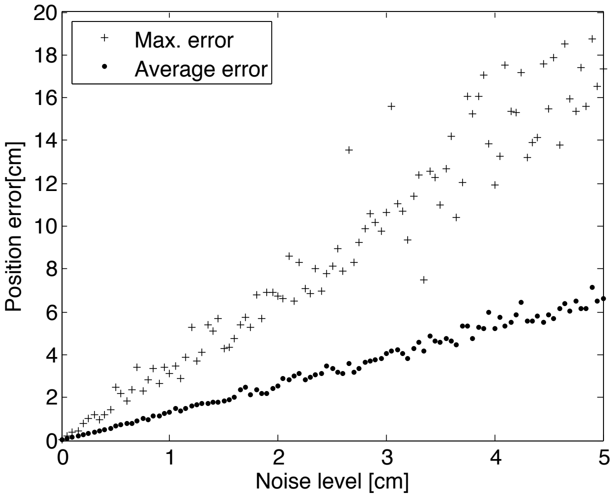

Fig. 5. Average and maximum localization errors for noise measurements with a standard deviation between 0 and 5 cm

Figure 5 shows the localization errors (average and maximum deviation) with different values of noise levels. The average localization error presents an approximately linear relationship with the standard value of the noise level, being approximately 1.5 times the noise level. The maximum localization error presents also a linear relationship of approximately 3 times the noise levels in the 0 to 2.5 cm noise range. This results shows an excellent first estimation of the beacon node positions considering no initial estimation was used. These results can be used as an initial seed in optimization algorithms to improve further the node positions.

VII. Conclusions and future work

In this paper we presented a solution for the auto-localization problem of LPS systems. First the rigidity theory was used to obtain the precise conditions in which an auto-localization problem is solvable for LPS systems. Satisfying these conditions guarantees that we have all the necessary information to find all the nodes positions. We also proposed a new method to calculate the beacon nodes position. In contrast with the other available techniques, our method don't rely on a first estimation of the beacon positions. Since the method uses a linearization of the auto-localization problem, it doesn't presents local minima errors. The method was developed for a 2D LPS system and for a 3D case where all beacon nodes are positioned in a plane. The simulation results shows a good estimation of the beacons position and can be used as an initial value for other methods to improve the estimation.

We plan to test the proposed algorithm in a real scenario with a 3D ultrasonic positioning system. We also want to improve the algorithm in order to be robust against the presence of outliers, which is a well known problem in ultrasonic systems. Finally we will explore ways to eliminate some of the restriction of the 3D algorithm, e.g. not requiring that all the beacon nodes must be positioned in a plane.

VIII. Future work

References

[1] A. Mahajan and F. Figueroa, "An automatic self-installation and calibration method for a 3d position sensing system using ultrasonics." *Robotics and Autonomous Systems*, vol. 28, no. 4, pp. 281–294, 1999.

[2] J. Prieto, A. Jimenez, J. Guevara, J. Ealo, F. Seco, J. Roa, and F. Ramos, "Performance evaluation of 3d-locus advanced acoustic lps," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 8, pp. 2385 –2395, aug. 2009.

[3] F. D. Ruiz, J. Ureña, J. M. Villadangos, I. Gude, J. J. García, Á. Hernández, and A. Jiménez, "Optimal test-point positions for calibrating an ultrasonic lps system." in *ETFA*. IEEE, 2008, pp. 338–344.

[4] A. Nishitani, Y. Nishida, T. Hori, and H. Mizoguchi, "Portable ultrasonic 3d tag system based on a quick calibration method," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 2004.

[5] P. Duff and H. Muller, "Autocalibration algorithm for ultrasonic location systems," in *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*. Washington, DC, USA: IEEE Computer Society, 2003, p. 62.

[6] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," in *Proceedings of Autonomous Underwater Vehicles, 2004*, 2004.

[7] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *IEEE INFOCOM*, Miami, FL, March 2005.

[8] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides, and B. Anderson, "Network localization in partially localizable networks," in *Proceedings of IEEE INFOCOM 2005*, Miami, FL, Mar. 2005.

[9] T. Eren, D. K. Goldenberg, W. Whiteley, and Y. R. Yang, "Rigidity, computation, and randomization in network localization," in *In Proceedings of IEEE INFOCOM 04, Hong Kong*, 2004, pp. 2673–2684.

[10] R. Mautz and W. Ochieng, "A robust indoor positioning and auto-localisation algorithm," *Journal of Global Positioning Systems*, vol. 6, no. 1, pp. 38–46, 2007.

[11] 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," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1663–1678, Dec. 2006.

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

[13] A. E. Varvitsiotis, "Algebraic and combinatorial techniques in rigidity theory," Master's thesis, National and Kapodistrian University of Athens, Greece, July 2009.

[14] T. Eren, B. D. O. Anderson, W. Whiteley, A. S. Morse, and P. N. Belhumeur, "Merging globally rigid formations of mobile autonomous agents," in *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. IEEE Computer Society, 2004, pp. 1260–1261.

[15] S. J. Gortler, A. D. Healy, and D. P. Thurston, "Characterizing generic global rigidity," 2007. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:0710.0926

[16] G. Mao, B. Fidan, G. Mao, and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2009.

[17] R. Connelly, "Generic global rigidity," *Discrete & Computational Geometry*, vol. 33, no. 4, pp. 549–563, 2005.