Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization

Pratik Biswas
Dept. of Electrical Engineering
Stanford University
CA 94305
pbiswas@stanford.edu

Yinyu Ye
Dept of Management Science and Engineering
and by courtesy, Electrical Engineering
Stanford University
Stanford, CA 94305
yinyu-ye@stanford.edu

ABSTRACT

We describe an SDP relaxation based method for the position estimation problem in wireless sensor networks. The optimization problem is set up so as to minimize the error in sensor positions to fit distance measures. Observable gauges are developed to check the quality of the point estimation of sensors or to detect erroneous sensors. The performance of this technique is highly satisfactory compared to other techniques. Very few anchor nodes are required to accurately estimate the position of all the unknown nodes in a network. Also the estimation errors are minimal even when the anchor nodes are not suitably placed within the network or the distance measurements are noisy.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Convex programming
; G.4 [**Mathematical Software**]: Algorithm design and analysis

General Terms

Algorithms

Keywords

Semidefinite Programming, Sensor Network Localization

1. INTRODUCTION

There has been an increase in the use of ad hoc wireless sensor networks for monitoring environmental information (temperature, sound levels, light etc) across an entire physical space. Typical networks of this type consist of a large number of densely deployed sensor nodes which must gather local data and communicate with other nodes. The sensor data from these nodes are relevant only if we know what location they refer to. Therefore knowledge of the node positions becomes imperative. The use of a GPS system is a very expensive solution to this requirement.

Instead, techniques to estimate node positions are being developed that rely just on the measurements of distances between neighboring nodes. The distance information could be based on criterion like time of arrival, time-difference of arrival and received signal strength. Depending on the accuracy of these measurements and processor, power and memory constraints at each of the nodes, there is some degree of error in the distance information. Furthermore, it is assumed that we already know the positions of a few anchor nodes. The problem of finding the positions of all the nodes given a few anchor nodes and relative distance information between the nodes is called the position estimation or localization problem.

This report describes an SDP relaxation based method for the position estimation problem in sensor networks. The optimization problem is set up so as to minimize the error in sensor positions for fitting the distance measures. Observable gauges are developed to measure the quality of the distance data. The basic idea behind the technique is to convert the nonconvex quadratic distance constraints into linear constraints by introducing a relaxation to remove the quadratic term in the formulation. Similar relaxations were developed for solving other distance geometry problems, see, e.g., Alfakih et al. [1] and Laurent [11].

The performance of this technique is highly satisfactory compared to other techniques. Very few anchor nodes are required to accurately estimate the position of all the unknown nodes in a network. Also the estimation errors are minimal even when the anchor nodes are not suitably placed within the network.

The next section gives a brief overview of the related work in this area. Then a detailed description of the SDP relaxation technique is provided and analyzed. Finally, the results of this technique for a few sensor network configurations are presented, and future research work and directions are proposed.

2. RELATED APPROACHES

It will be helpful to first introduce some notations. The trace of a given matrix A, denoted by $\mathrm{Trace}(A)$, is the sum of the entries on the main diagonal of A. We use I, e and $\mathbf{0}$ to denote the identity matrix, the vector of all ones and the vector of all zeros, whose dimension will be clear in the

IPSN'04, April 26–27, 2004, Berkeley, California, USA.

context. The inner product of two vector p and q is denoted by $\langle p, q\rangle$. The 2-norm of a vector x, denoted by $\|x\|$, is defined by $\sqrt{\langle x, x\rangle}$. A positive semidefinite matrix X is represented by $X \succeq 0$.

A great deal of research has been done on the topic of position estimation in ad-hoc networks ([8, 9]). Most techniques use distance or angle measurements from a fixed set of reference or anchor nodes, see [7, 13, 14, 15, 16, 17]; or employ a grid of beacon nodes with known positions, see [6, 10]. Also see Moré and Wu [12] for solving another distance geometry problem.

One closely related approach is described by Doherty et al. [7] wherein the proximity constraints between nodes which are within 'hearing distance' of each other are modeled as convex constraints. Then a feasibility problem can be solved by efficient convex programming techniques. Suppose 2 nodes x_1 and x_2 are within radio range R of each other, the proximity constraint can be represented as a convex second order cone constraint of the form

$$\|x_1 - x_2\|_2 \leq R \tag{1}$$

This can be formulated as a matrix linear inequality (Boyd et al. [5]):

$$\begin{bmatrix} I_2 R & x_1 - x_2 \\ (x_1 - x_2)^T & R \end{bmatrix} \succeq 0 \tag{2}$$

Alternatively, if the exact distance $r_{1,2} \leq R$ is known, we could set the constraint

$$\|(x_1 - x_2)\|_2 \leq r_{1,2}. \tag{3}$$

The second-order cone method for solving Euclidean metric problems can be also found in Xue and Ye [19] where its superior polynomial complexity efficiency is presented.

However, this technique yields good results only if the anchor nodes are placed on the outer boundary, since the estimated positions of their convex optimization model all lie within the convex hull of the anchor nodes. So if the anchor nodes are placed to the interior of the network, the position estimation of the unknown nodes will also tend to the interior, yielding highly inaccurate results. For example, with just 5 anchors in a random 200 node network, the estimation error is almost twice the radio range.

One may ask why not add, if $r_{1,2}$ is known, another "bounding away" constraint

$$\|(x_1 - x_2)\|_2 \geq r_{1,2}. \tag{4}$$

These two constraints are much tighter and would yield more accurate results. The problem is that the latter is not a convex constraint, so that the efficient convex optimization techniques cannot apply. The SDP relaxation method presented in this paper attempts to formulate tighter convex constraints similar to 4. Then we relax it to linear matrix inequalities similar to (2). In fact, just 3 anchors are enough to achieve accurate results for the entire network.

Shang et al. [17] demonstrate the use of a data analysis technique called "multidimensional scaling" (MDS) in estimating positions of unknown nodes. Firstly, using basic connectivity or distance information, a rough estimate of relative node distances is made. Then MDS is used to obtain a relative map of the node positions. Finally an absolute map is obtained by using the known node positions. This technique works well with few anchors and reasonably high connectivity. For instance, for a connectivity level of 12 and 2% anchors, the error is about half of the radio range.

The techniques described above are predominantly centralized although distributed versions can be developed. The available distance information between all the nodes must be present on a single computer for these techniques to work. The distributed approach has the advantage that the techniques can be executed on the sensor nodes themselves thus removing the need to relay all the information to a central computer. Many techniques have been proposed that try to emphasize the ad-hoc nature of computation required in them.

Niculescu and Nath [13] describe the "DV-Hop" approach which is quite effective in dense and regular topologies. The anchor nodes flood their position information to all the nodes in the network. Each node then estimates its own position by performing a triangulation to three or more anchors. For more irregular topologies however, the accuracy can deteriorate to the radio range.

Savarese et al. [14] present a 2 phase algorithm in which the start-up phase involves finding the rough positions of the nodes using a technique similar to the "DV-Hop" approach. The refinement phase improves the accuracy of the estimated positions by performing least squares triangulations using its own estimates and the estimates of the nodes in its own neighborhood. This method can accurately estimate points within one third of the radio range.

When the number of anchor nodes is high, the "iterative multiplication" technique proposed by Savvides et al. [15] yields good results. Nodes that are connected to 3 or more anchors compute their position by triangulation and upgrade themselves to anchor nodes. Now their position information can also be used by the other unknown nodes for their position estimation in the next iteration.

3. SEMIDEFINITE PROGRAMMING METHODS

We first present quadratic formulations of the position estimation problem, then introduce their semidefinite programming (SDP) models. For simplicity, let the sensor points be placed on a plane. Suppose we have m known points (anchors) $a_k \in \mathcal{R}^2$, $k = 1, ..., m$, and n unknown points (sensors) $x_j \in \mathcal{R}^2$, $j = 1, ..., n$. For a pair of two points in N_e, we have a Euclidean distance measure $\hat{d}_{kj}$ between a_k and x_j or $\hat{d}_{ij}$ between x_i and x_j; and for a pair of two points in N_l, we have a distance lower bound $\underline{r}_{kj}$ between a_k and x_j or $\underline{r}_{ij}$ between x_i and x_j; and for a pair of two points in N_u, we have a distance upper bound $\bar{r}_{kj}$ between a_k and x_j or $\bar{r}_{ij}$ between x_i and x_j. Then, the localization problem is to find x_js such that

$$\begin{array}{l} \|x_i - x_j\|^2 = (\hat{d}_{ij})^2, \ \|a_k - x_j\|^2 = (\hat{d}_{kj})^2, \ \forall (i,j), (k,j) \in N_e \\ \|x_i - x_j\|^2 \geq (\underline{r}_{ij})^2, \ \|a_k - x_j\|^2 \geq (\underline{r}_{kj})^2, \ \forall (i,j), (k,j) \in N_l \\ \|x_i - x_j\|^2 \leq (\bar{r}_{ij})^2, \ \|a_k - x_j\|^2 \leq (\bar{r}_{kj})^2, \ \forall (i,j), (k,j) \in N_u. \end{array}$$

Since these measures and bounds are typically noisy, a "softer" model would be to choose x_js such that the sum of errors is minimized:

$$\begin{array}{ll} \min & \sum_{i,j\in N_e,\ i<j} |\|x_i - x_j\|^2 - (\hat{d}_{ij})^2| \\ & + \sum_{k,j\in N_e} |\|a_k - x_j\|^2 - (\hat{d}_{kj})^2| \\ & + \sum_{i,j\in N_l,\ i<j} (\|x_i - x_j\|^2 - (\underline{r}_{ij})^2)_- \\ & + \sum_{k,j\in N_l} (\|a_k - x_j\|^2 - (\underline{r}_{kj})^2)_- \\ & + \sum_{i,j\in N_u,\ i<j} (\|x_i - x_j\|^2 - (\bar{r}_{ij})^2)_+ \\ & + \sum_{k,j\in N_u} (\|a_k - x_j\|^2 - (\bar{r}_{kj})^2)_+, \end{array}$$

where $(u)_-$ and $(u)_+$ are defined as

$$(u)_- = \max\{0, -u\} \quad \text{and} \quad (u)_+ = \max\{0, u\}.$$

Note that if each of these errors is squared, the minimization problem becomes a least squares problem. In this study, we choose minimizing the sum of absolute errors to show how the SDP model works.

By introducing slack variables αs and βs, the softer error minimization problem can be rewritten as

$$\begin{array}{ll} \min & \sum_{i,j\in N_e,\ i<j} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{k,j\in N_e} (\alpha_{kj}^+ + \alpha_{kj}^-) \\ & + \sum_{i,j\in N_l,\ i<j} \beta_{ij}^- + \sum_{k,j\in N_l} \beta_{kj}^- \\ & + \sum_{i,j\in N_u,\ i<j} \beta_{ij}^+ + \sum_{k,j\in N_u} \beta_{kj}^+ \\ \text{s.t.} & \|x_i - x_j\|^2 - (\hat{d}_{ij})^2 = \alpha_{ij}^+ - \alpha_{ij}^-,\ \forall\, i,j \in N_e,\ i<j, \\ & \|a_k - x_j\|^2 - (\hat{d}_{kj})^2 = \alpha_{kj}^+ - \alpha_{kj}^-,\ \forall\, k,j \in N_e, \\ & \|x_i - x_j\|^2 - (\underline{r}_{ij})^2 \geq -\beta_{ij}^-,\ \forall\, i,j \in N_l,\ i<j, \\ & \|a_k - x_j\|^2 - (\underline{r}_{kj})^2 \geq -\beta_{kj}^-,\ \forall\, k,j \in N_l, \\ & \|x_i - x_j\|^2 - (\underline{r}_{ij})^2 \leq \beta_{ij}^+,\ \forall\, i,j \in N_u,\ i<j, \\ & \|a_k - x_j\|^2 - (\underline{r}_{kj})^2 \leq \beta_{kj}^+,\ \forall\, k,j \in N_u \\ & \alpha_{ij}^+,\ \alpha_{ij}^-,\ \alpha_{kj}^+,\ \alpha_{kj}^-,\ \beta_{ij}^-,\ \beta_{kj}^-, \beta_{ij}^+,\ \beta_{kj}^+ \geq 0. \end{array} \tag{5}$$

Let $X = [x_1\ x_2\ ...\ x_n]$ be the $2 \times n$ matrix that needs to be determined. Then

$$\|x_i - x_j\|^2 = e_{ij}^T X^T X e_{ij},$$

$$\|a_i - x_j\|^2 = (a_i; e_j)^T [I \quad X]^T [I \quad X](a_i; e_j),$$

where e_{ij} is the vector with 1 at the ith position, -1 at the jth position and zero everywhere else; and e_j is the vector of all zero except -1 at the jth position. Let $Y = X^T X$. Then problem (5) can be rewritten as:

$$\begin{array}{ll} \min & \sum_{i,j\in N_e,\ i<j} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{k,j\in N_e} (\alpha_{kj}^+ + \alpha_{kj}^-) \\ & \sum_{i,j\in N_l,\ i<j} \beta_{ij}^- + \sum_{k,j\in N_l} \beta_{kj}^- \\ & \sum_{i,j\in N_u,\ i<j} \beta_{ij}^+ + \sum_{k,j\in N_u} \beta_{kj}^+ \\ \text{s.t.} & e_{ij}^T Y e_{ij} - (\hat{d}_{ij})^2 = \alpha_{ij}^+ - \alpha_{ij}^-,\ \forall\, i,j \in N_e,\ i<j, \\ & (a_k; e_j)^T \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} (a_k; e_j) - (\hat{d}_{kj})^2 = \alpha_{kj}^+ - \alpha_{kj}^-, \\ & \forall\, k,j \in N_e, \\ & e_{ij}^T Y e_{ij} - (\underline{r}_{ij})^2 \geq -\beta_{ij}^-,\ \forall\, i,j \in N_l,\ i<j, \\ & (a_k; e_j)^T \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} (a_k; e_j) - (\underline{r}_{kj})^2 \geq -\beta_{kj}^-, \\ & \forall\, k,j \in N_l, \\ & e_{ij}^T Y e_{ij} - (\underline{r}_{ij})^2 \leq \beta_{ij}^+,\ \forall\, i,j \in N_u,\ i<j, \\ & (a_k; e_j)^T \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} (a_k; e_j) - (\underline{r}_{kj})^2 \leq \beta_{kj}^+, \\ & \forall\, k,j \in N_u, \\ & \alpha_{ij}^+,\ \alpha_{ij}^-,\ \alpha_{kj}^+,\ \alpha_{kj}^-,\ \beta_{ij}^-,\ \beta_{kj}^-, \beta_{ij}^+,\ \beta_{kj}^+ \geq 0, \\ & Y = X^T X. \end{array} \tag{6}$$

Unfortunately, the above problem is not a convex optimization problem. Doherty et al. [7] essentially ignore the non-convex inequality and equality constraints but keep the convex ones, resulting in a convex second-order cone optimization problem, where all position estimations only lie in the convex hull of anchors. Others have essentially used various types of nonlinear equation and global optimization solvers to solve similar quadratic models, where final solutions are highly dependent on initial solutions and search directions.

Our method is to relax problem (6) to a semidefinite program: Change $Y = X^T X$ in (6) to $Y \succeq X^T X$. This matrix inequality is equivalent to (e.g., Boyd et al. [5])

$$Z := \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq 0.$$

Then, the problem can be written as a standard SDP problem:

$$\begin{array}{ll} \min & \sum_{i,j\in N_e,\ i<j} (\alpha_{ij}^+ + \alpha_{ij}^-) + \sum_{k,j\in N_e} (\alpha_{kj}^+ + \alpha_{kj}^-) \\ & \sum_{i,j\in N_l,\ i<j} \beta_{ij}^- + \sum_{k,j\in N_l} \beta_{kj}^- \\ & \sum_{i,j\in N_u,\ i<j} \beta_{ij}^+ + \sum_{k,j\in N_u} \beta_{kj}^+ \\ \text{s.t.} & (1;0;\mathbf{0})^T Z (1;0;\mathbf{0}) = 1 \\ & (0;1;\mathbf{0})^T Z (0;1;\mathbf{0}) = 1 \\ & (1;1;\mathbf{0})^T Z (1;1;\mathbf{0}) = 2 \\ & (\mathbf{0}; e_{ij})^T Z (\mathbf{0}; e_{ij}) - \alpha_{ij}^+ + \alpha_{ij}^- = (\hat{d}_{ij})^2, \\ & \forall\, i,j \in N_e,\ i<j, \\ & (a_k; e_j)^T Z (a_k; e_j) - \alpha_{kj}^+ + \alpha_{kj}^- = (\hat{d}_{kj})^2,\ \forall\, k,j \in N_e, \\ & (\mathbf{0}; e_{ij})^T Z (\mathbf{0}; e_{ij}) + \beta_{ij}^- \geq (\underline{r}_{ij})^2,\ \forall\, i,j \in N_l,\ i<j, \\ & (a_k; e_j)^T Z (a_k; e_j) + \beta_{kj}^- \geq (\underline{r}_{kj})^2,\ \forall\, k,j \in N_l, \\ & (\mathbf{0}; e_{ij})^T Z (\mathbf{0}; e_{ij}) - \beta_{ij}^+ \leq (\underline{r}_{ij})^2,\ \forall\, i,j \in N_u,\ i<j, \\ & (a_k; e_j)^T Z (a_k; e_j) - \beta_{kj}^+ \leq (\underline{r}_{kj})^2,\ \forall\, k,j \in N_u, \\ & \alpha_{ij}^+,\ \alpha_{ij}^-,\ \alpha_{kj}^+,\ \alpha_{kj}^-,\ \beta_{ij}^-,\ \beta_{kj}^-, \beta_{ij}^+,\ \beta_{kj}^+ \geq 0, \\ & Z \succeq 0. \end{array} \tag{7}$$

A worst-case complexity result to solve the SDP relaxation can be derived from employing interior-point algorithms, e.g., Benson et al. [2].

THEOREM 1. *Let $k = 3 + |N_e| + |N_l| + |N_u|$, the number of constraints. Then, the worst-case number of total arithmetic operations to compute an ϵ-solution of (7), meaning its objective value is at most $\epsilon (> 0)$ above the minimal one, is bounded by $O(\sqrt{n+k}(n^3 + n^2 k + k^3) \log \frac{1}{\epsilon})$, in which $\sqrt{n+k} \log \frac{1}{\epsilon}$ represents the bound on the worst-case number of interior-point algorithm iterations.*

Practically, the number of interior-point algorithm iterations to compute a fairly accurate solution is a constant, $20-30$, for semidefinite programming, and k is in the range of $O(n)$. Thus, the number of operations is typically bounded by $O(n^3)$ in solving a localization problem with n sensors.

4. SDP MODEL ANALYSES

The matrix of Z of (7) has $2n + n(n+1)/2$ unknown variables. Consider the case that among $\{k, i, j\}$, there are $2n + n(n+1)/2$ of the pairs in N_e, and the objective has zero value for the minimal solution of (7). Then we have at least $2n + n(n+1)/2$ linear equalities among the constraints. Moreover, if these equalities are linearly independent, then Z has a unique solution. Therefore, we can show

PROPOSITION 1. *If there are $2n + n(n+1)/2$ distance pairs each of which has an accurate distance measure and other distance bounds are feasible. Then, the minimal value* *of (7) is zero. Moreover, if (7) has a unique minimal solution*

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix},$$

then we must have $\bar{Y} = (\bar{X})^T\bar{X}$ and $\bar{X}$ equal true positions of the unknown sensors. That is, the SDP relaxation solves the original problem exactly.

PROOF. Let X^* be the true locations of the n points, and

$$Z^* = \begin{pmatrix} I & X^* \\ (X^*)^T & (X^*)^T X^* \end{pmatrix}.$$

Then Z^* and all slack variables being zero is a feasible solution for (7).

On the other hand, since $\bar{Z}$ is the unique solution to satisfy the $2n+n(n+1)/2$ equalities, we must have $\bar{Z} = Z^*$ so that $\bar{Y} = (X^*)^T X^* = \bar{X}^T\bar{X}$. $\square$

We present a simple case to show what it means for the system has a unique solution. Consider $n = 1$ and $m = 3$. The accurate distance measures from unknown b_1 to known a_1, a_2 and a_3 are d_{11}, d_{21} and d_{31}, respectively. Therefore, the three linear equations are

$$\begin{aligned} y - 2x^T a_1 &= (d_{11})^2 - \|a_1\|^2 \\ y - 2x^T a_2 &= (d_{21})^2 - \|a_2\|^2 \\ y - 2x^T a_3 &= (d_{31})^2 - \|a_3\|^2 \end{aligned}$$

This system has a unique solution if it has a solution and the matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ a_1 & a_2 & a_3 \end{pmatrix}$$

is nonsingular. This essentially means that the three points a_1, a_2 and a_3 are not on the same line, and then $\bar{x} = b_1$ can be uniquely determined. Here, the SDP method reduces to the so-called triangular method. Proposition 1 and the example show that the SDP relaxation method has the advantage of the triangular method in solving the original problem.

5. PROBABILISTIC AND ERROR ANALYSES

The case discussed in Proposition 1 is deterministic. Alternatively, each x_j can be viewed a random point $\tilde{x}_j$ since the distance measures contain random errors. Then the solution to the SDP problem provides the first and second moment information on $\tilde{x}_j$, $j = 1, ..., n$. Such an interpretation appears to be first stated in Bertsimas and Ye [3].

Generally, we have

$$\mathrm{E}[\tilde{x}_j] \sim \bar{x}_j, \quad j = 1, ..., n$$

and

$$\mathrm{E}[\tilde{x}_i^T\tilde{x}_j] \sim \bar{Y}_{ij}, \quad i, j = 1, ..., n.$$

where

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

is the optimal solution of the SDP problem.

Thus,

$$\bar{Y} - \bar{X}^T\bar{X}$$

represents the co-variance matrix of $\tilde{x}_j$, $j = 1, ..., n$.

These quantities also constitute error management and analyses of the original problem data. For example,

$$\mathrm{Trace}(\bar{Y} - \bar{X}^T\bar{X}) = \sum_{j=1}^{n}(\bar{Y}_{jj} - \|\bar{x}_j\|^2),$$

the total trace of the co-variance matrix, measures the quality of distance sample data d_{ij} and d_{kj}. In particular, individual trace

$$\bar{Y}_{jj} - \|\bar{x}_j\|^2,$$

which is also the variance of $\|\tilde{x}_j\|$, helps us to detect possible distance measure errors, and outlier or defect sensors. These errors often occur in real applications either due to the lack of data information or noisy measurement, and are often difficult to detect since the true location of sensors is unknown.

We again use the same simple case to illustrate our theory. Consider $n = 1$ and $m = 3$. The inexact distance measures from unknown b_1 to known a_1, a_2 and a_3 are $d_{11}+\epsilon$, $d_{21}+\epsilon$ and $d_{31}+\epsilon$, respectively, where ϵ is a random error with zero mean. Therefore, the three linear equations are

$$\begin{aligned} \bar{y} - 2\bar{x}^T a_1 + \|a_1\|^2 &= (d_{11})^2 + 2\epsilon d_{11} + \epsilon^2 \\ \bar{y} - 2\bar{x}^T a_2 + \|a_2\|^2 &= (d_{21})^2 + 2\epsilon d_{21} + \epsilon^2 \\ \bar{y} - 2\bar{x}^T a_3 + \|a_3\|^2 &= (d_{31})^2 + 2\epsilon d_{31} + \epsilon^2. \end{aligned}$$

Taking expect values on both sides, we have

$$\begin{aligned} \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_1 + \|a_1\|^2 &= (d_{11})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_2 + \|a_2\|^2 &= (d_{21})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - 2\mathrm{E}[\bar{x}]^T a_3 + \|a_3\|^2 &= (d_{31})^2 + \mathrm{E}[\epsilon^2] \end{aligned}$$

or

$$\begin{aligned} \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T\mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_1\|^2 &= (d_{11})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T\mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_2\|^2 &= (d_{21})^2 + \mathrm{E}[\epsilon^2] \\ \mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T\mathrm{E}[\bar{x}] + \|\mathrm{E}[\bar{x}] - a_3\|^2 &= (d_{31})^2 + \mathrm{E}[\epsilon^2]. \end{aligned}$$

The solution to the linear equation is

$$\mathrm{E}[\bar{x}] = b_1,$$

and

$$\mathrm{E}[\bar{y}] - \mathrm{E}[\bar{x}]^T\mathrm{E}[\bar{x}] = \mathrm{E}[\epsilon^2]$$

or

$$\mathrm{E}[\bar{y}] = \mathrm{E}[\epsilon^2] + \|b_1\|^2.$$

That is, $\bar{x}$ is a point estimate of b_1. Moreover, from

$$\mathrm{E}[\bar{y}] - \mathrm{E}[\|\bar{x}\|^2] + \mathrm{E}[\|\bar{x}\|^2] - \|b_1\|^2 = \mathrm{E}[\bar{y}] - \|b_1\|^2 = \mathrm{E}[\epsilon^2],$$

we have

$$\mathrm{E}[\bar{y} - \|\bar{x}\|^2] \le \mathrm{E}[\epsilon^2] \quad \text{and} \quad \mathrm{E}[\|\bar{x}\|^2] - \|b_1\|^2 \le \mathrm{E}[\epsilon^2],$$

so that the quantity of $y - \|\bar{x}\|^2$ is a lower bound estimate for the error variance and the variance of $\bar{x}$ is also bounded by the error variance. This quantity gives an interval estimation of b_1.

More generally, we have

Proposition 2. *Let the noisy measurements*

$$\hat{d}_{ij} = d_{ij} + \epsilon_i + \epsilon_j, \ \forall\, i \neq j$$

and

$$\hat{d}_{kj} = d_{kj} + \epsilon_j, \ \forall k, j$$

where d_{ij} are the true distances and ϵ_j are independent random errors with zero mean. Moreover, let the minimal value be zero in (7) and the anchor points are linear independent. Then, we have

$$E[\bar{x}_j] = b_j \quad \textit{and} \quad E[\bar{Y}_{jj}] = \|b_j\|^2 + E[\epsilon_j^2] \ \forall\, j$$

and

$$E[\bar{Y}_{ij}] = (b_i)^T b_j \ \forall\, i \neq j,$$

where b_j is the true position of x_j, $j = 1, ..., n$, and

$$\bar{Z} = \begin{pmatrix} I & \bar{X} \\ \bar{X}^T & \bar{Y} \end{pmatrix}$$

is the minimizer of (7).

Proof. We have, for all i, j, k,

$$\bar{Y}_{ii} - 2\bar{Y}_{ij} + \bar{Y}_{jj} = (d_{ij} + \epsilon_i + \epsilon_j)^2$$

$$\bar{Y}_{jj} - 2\bar{x}_j^T a_k + \|a_k\|^2 = (d_{kj} + \epsilon_j)^2.$$

Taking expect values on both sides, we have

$$\mathrm{E}[\bar{Y}_{ii}] - 2\mathrm{E}[\bar{Y}_{ij}] + \mathrm{E}[\bar{Y}_{jj}] = (d_{ij})^2 + \mathrm{E}[\epsilon_i^2] + \mathrm{E}[\epsilon_j^2]$$

and

$$\mathrm{E}[\bar{Y}_{jj}] - \mathrm{E}[\bar{x}_j]^T \mathrm{E}[\bar{x}_j] + \|\mathrm{E}[\bar{x}_j] - a_k\|^2 = (d_{kj})^2 + \mathrm{E}[\epsilon_j^2],$$

or

$$\mathrm{E}[\bar{Y}_{ii}] - 2\mathrm{E}[\bar{Y}_{ij}] + \mathrm{E}[\bar{Y}_{jj}] - \|\mathrm{E}[\bar{x}_i] - \mathrm{E}[\bar{x}_j]\|^2 + \|\mathrm{E}[\bar{x}_j] - \mathrm{E}[\bar{x}_i]\|^2$$

$$= (d_{ij})^2 + \mathrm{E}[\epsilon_i^2] + \mathrm{E}[\epsilon_j^2]$$

and

$$\mathrm{E}[\bar{Y}_{jj}] - \mathrm{E}[\bar{x}_j]^T \mathrm{E}[\bar{x}_j] + \|\mathrm{E}[\bar{x}_j] - a_k\|^2 = (d_{kj})^2 + \mathrm{E}[\epsilon_j^2].$$

Thus,

$$\mathrm{E}[\bar{x}_j] = b_j \quad \text{and} \quad \mathrm{E}[\bar{Y}_{jj}] = \|b_j\|^2 + \mathrm{E}[\epsilon_j^2] \ \forall\, j$$

and

$$\mathrm{E}[\bar{Y}_{ij}] = (b_i)^T b_j \ \forall\, i \neq j$$

is the solution satisfying these equations. $\square$

6. COMPUTATIONAL RESULTS

Simulations were performed on a network of 50 sensors or nodes randomly placed in a square region of size 1×1 centered at the origin. The distances between the nodes was calculated. If the distance between 2 notes was less than a given *radiorange* between $[0, 1]$, a random error was added to it

$$\hat{d}_{ij} = d_{ij} \cdot (1 + randn(1) * noisyfactor),$$

where *noisyfactor* was a given number between $[0, 1]$ and $randn(1)$ was a standard normal random variable. If the distance was beyond the given *radiorange*, only the lower bound constraint, $\geq$ *radiorange*, was applied if necessary. The selection of the lower bounding constraint was based on an iterative active-constraint generation technique. Because most of these "bounding away" constraints, i.e., the constraints between two remote nodes, would be inactive or redundant at an optimal solution, very few of these constraints were needed in computation.

The average estimation error and average trace were defined by

$$\frac{1}{n} \cdot \sum_{j=1}^{n} \|\bar{x}_j - a_j\| \quad \text{and} \quad \frac{1}{n} \cdot \mathrm{Trace}(\bar{Y} - \bar{X}^T \bar{X}),$$

respectively, where $\bar{x}_j$ comes from the SDP solution and a_j is the true position of the jth node. Connectivity indicates how many of the nodes, on average, are within the radio range of a node.

Also the original and the estimated sensors were plotted. The (blue) diamond nodes refer to the positions of the anchors; (green) circle nodes to the original locations, A, of the unknown sensors; and (red) asterisk nodes to their estimated positions from $\bar{X}$. The discrepancies in the positions can be estimated by the offsets between the original and the estimated points as indicated by the solid lines.

The effect of variable radio ranges and as a result, connectivity, was observed in Figure 1. The radio range was varied from 0.2 to 0.35. In Figure 1(b), for the four sensors with large error estimation, their individual traces made up most of the total traces, which match where the real errors are accurately, see Figure 2 for the correlation between individual error and trace for each unknown sensor for cases Figure 1(a) and Figure 1(b).

In comparison, for the same case Figure 1(c), we computed the results from the Doherty et al. [7] method with the number of anchors 10 and 25, and depicted their pictures in Figure 3. As we expected, the estimated positions were all in the convex hull of the anchors.

Next, we set the *noisyfactor* to 0.05 and radio range to 0.3, we increase the anchors to 7, and their pictures are illustrated in Figure 4. Here, each of the distance data has up to 5% error, either plus or minus. Again, one can see the erroneous estimations are reflected in their individual traces.

Then, we increase *noisyfactor* to 0.10 (that is, each of the distance data has up to 10% error either plus or minus), the number of anchors to 7, but vary the radio range in Figure 5. Even with 10% error measurement, the position estimation for the sensors near anchor nodes is still fairly accurate.

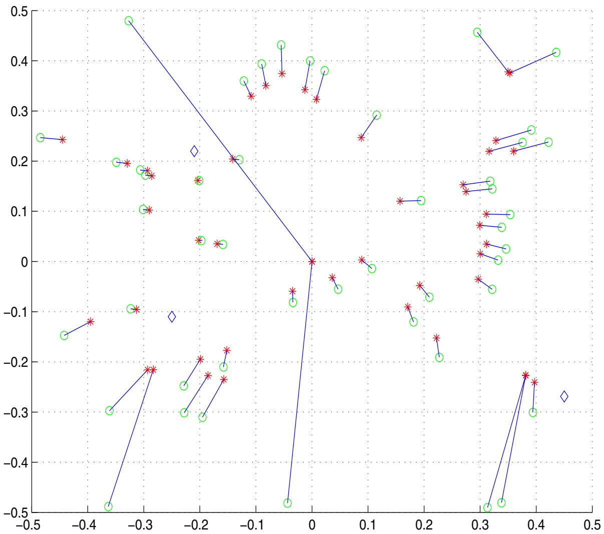
The computational results presented here were generated using the interior-point algorithm SDP solvers SeDuMi of [18] and DSDP2.0 of [2] with their interfaces to Matlab. The average time to solve one of the 50 sensor problem is 8 seconds by SeDuMi (with the primal formulation) or 2 seconds by DSDP2.0 on a Pentium 1.2 GHz and 500 MB PC. DSDP is faster due to the fact that the data structure of the problem is more suitable for DSDP.

7. WORK IN PROGRESS

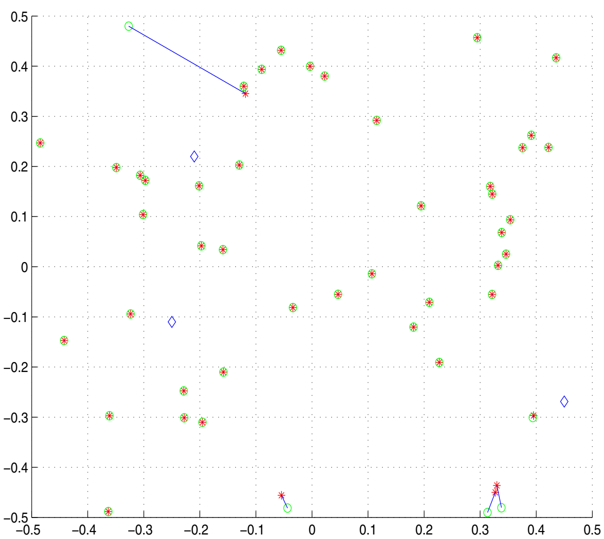
The current work is to improve computational efficiency for solving SDP problem (7) using distributed computing. Recall that our unknown matrix has a form

$$\begin{pmatrix} I & X \\ X^T & Y \end{pmatrix}$$

[error:0.08, connectivity:5.8, average trace:0.09, radio range=0.20.]



[error:0.0076, connectivity:7.8, average trace:0.0021, radio range=0.25.]



[error:1.8e-4, connectivity:10.5, average trace:4.3e-5, radio range=0.30.]

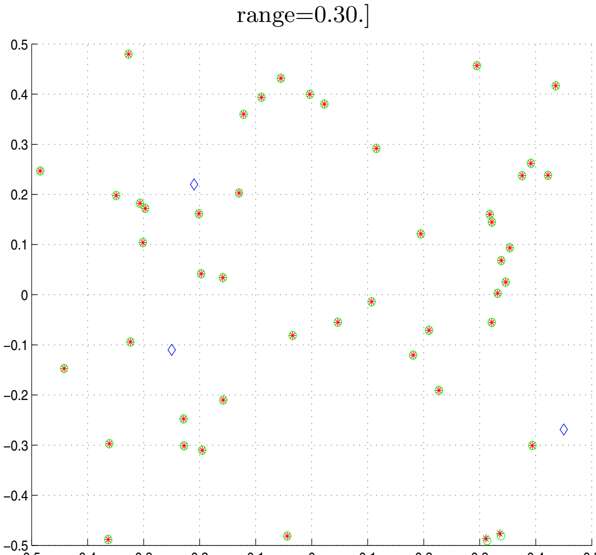
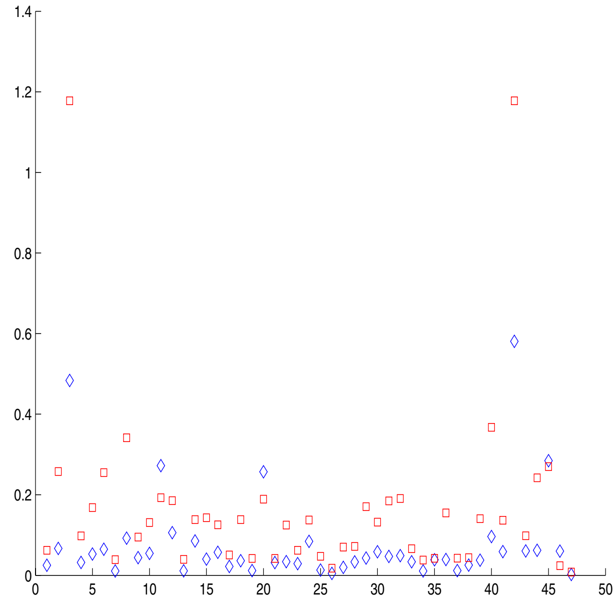


Figure 1: Position estimations with 3 anchors, noisy factor=0, and various radio ranges.

[Error and trace correlation in Figure 1(a).]



[Error and trace correlation in Figure 1(b).]

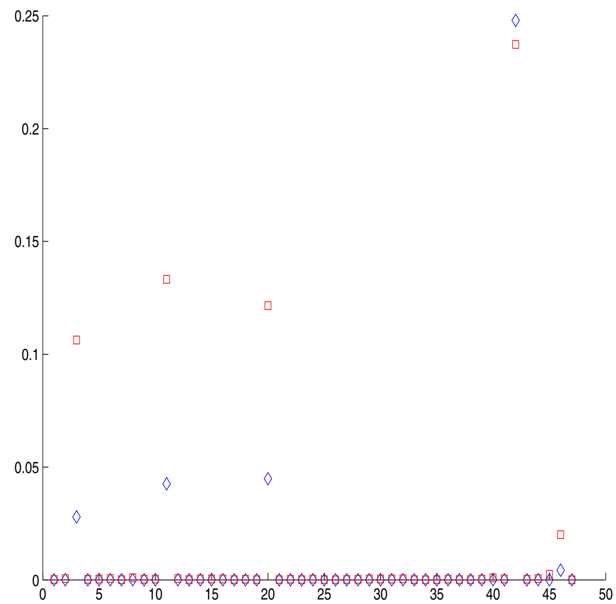


Figure 2: Diamond: the offset distance between estimated and true positions, Box: the square root of individual trace $\bar{Y}_{jj} - \|\bar{x}_j\|^2$.

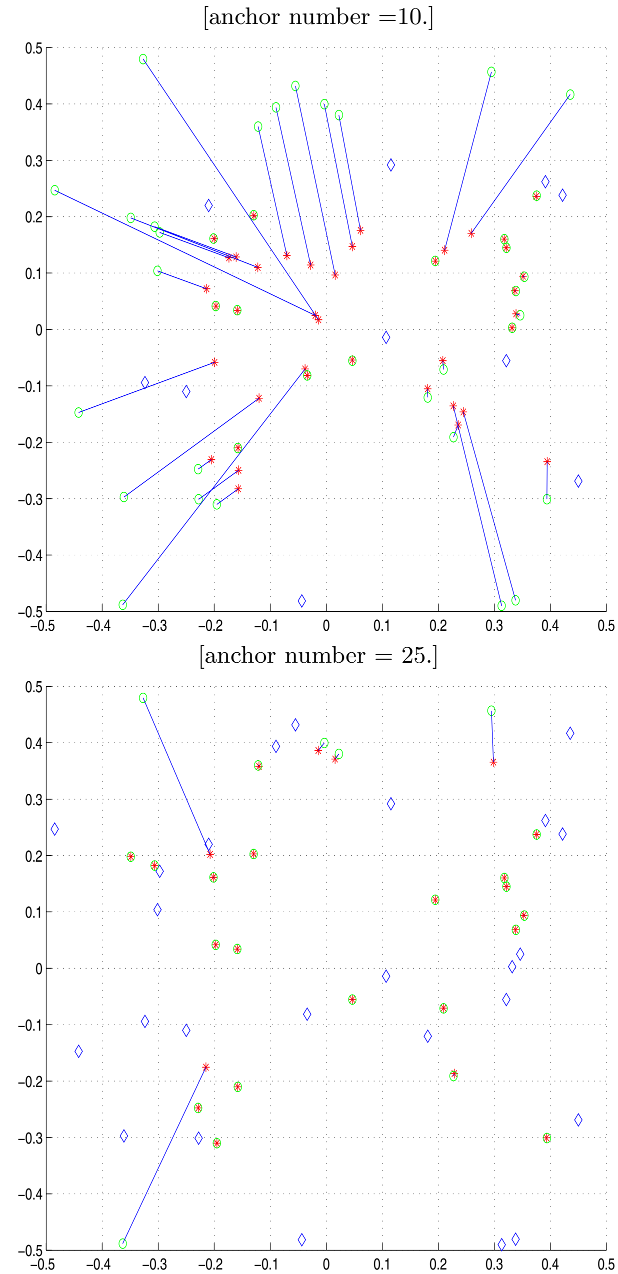


Figure 3: Position estimations by Doherty et al. [7], radio range=0.30, noisy factor=0, and various number of anchors.

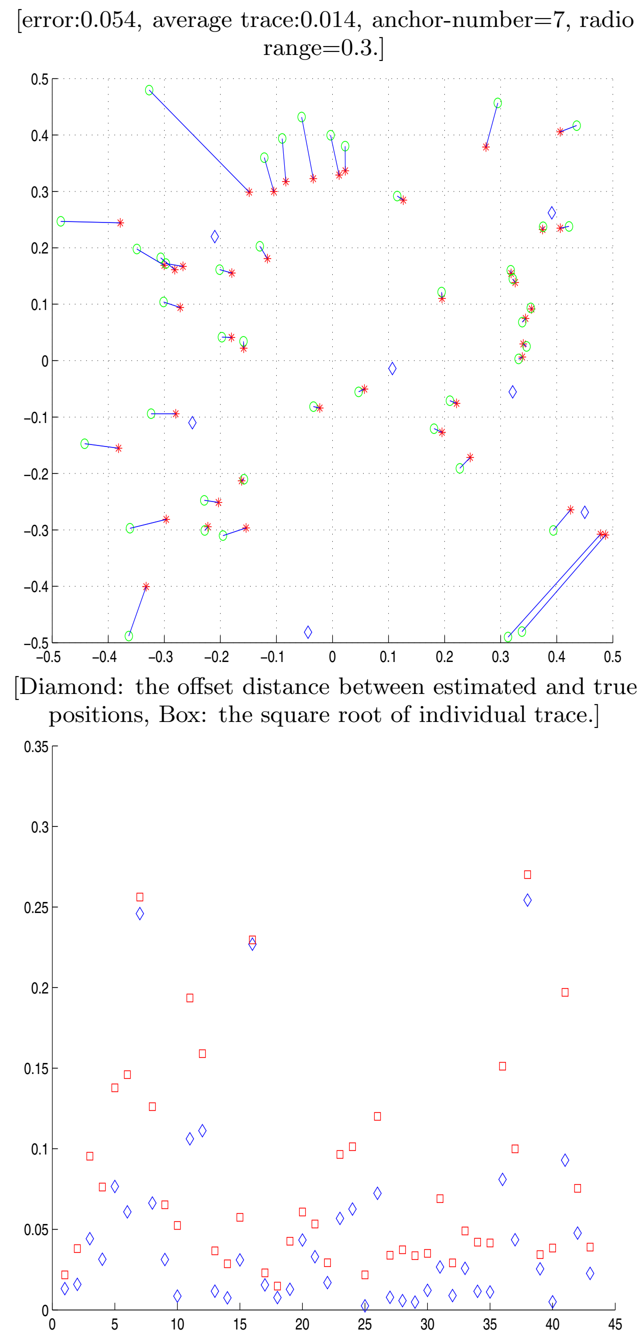


Figure 4: Position estimations and Error-Trace correlation with noisy factor=0.05.

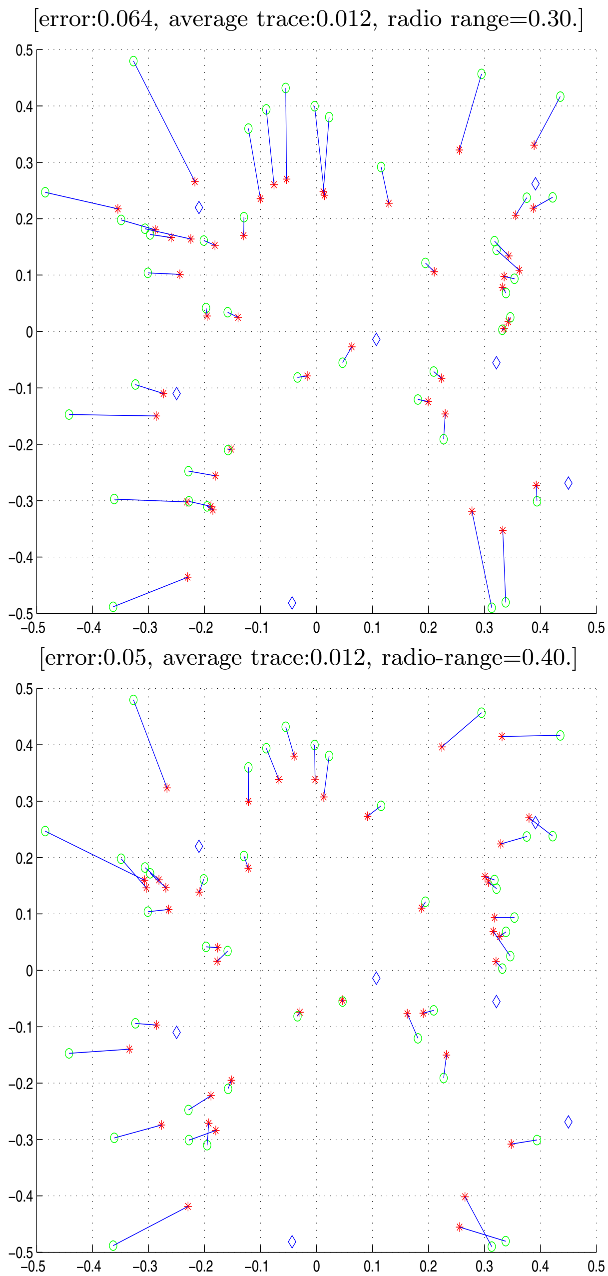


Figure 5: Position estimations with 7 anchors, noisy factor=0.1, and various radio range.

where it can be decomposed into K principle blocks

$$\begin{pmatrix} I & X_1 & X_2 & \dots & X_K \\ X_1^T & Y_{11} & Y_{12} & \dots & Y_{1K} \\ X_2^T & Y_{21} & Y_{22} & \dots & Y_{2K} \\ \dots & \dots & \dots & \dots & \dots \\ X_K^T & Y_{K1} & Y_{K2} & \dots & Y_{KK} \end{pmatrix}$$

where the kth principle block matrix is

$$\begin{pmatrix} I & X_k \\ X_k^T & Y_{kk} \end{pmatrix}$$

Then, we can solve the kth block problem, assuming the off diagonal blocks are fixed, in a distributed fashion for $k = 1, ..., K$. That is, ignoring other block's solutions, each of these problems can be solved locally and independently. Thereafter, we have new X_k and Y_{kk} for $k = 1, ..., K$, and Y_{ki} can be then updated to $X_i^T X_k$ among the blocks. We repeat this process iteratively till the trace of the solution converges.

The physical interpretation of this decomposition process is as follows. The entire sensor network is divided into smaller clusters based on geographical location and the algorithm is applied on each cluster independently. Based on the trace errors , we update some of the estimated unknown points to anchor status and repeat the estimation algorithm with the increased anchor density. This process is repeated until all the points in the network have been accurately estimated. For more details, the reader is referred to a preliminary report [4]. We depict some initial results in Figure 6, where 200 anchors and 2000 sensors were presented, and the radio range was 0.06. The problem was clustered into 64 blocks and it was solved on the same single machine in less than 2 minutes.

Also, in our simulations, the radio range connectivity model used is very simplistic and unrealistic. It may not be an accurate physical representation of the highly time and space varying communication ranges for the sensors. However, the SDP relaxation approach can be easily extended to more real life scenarios with more complicated connectivity and noise models. The approach only depends on sufficient pairwise distance information between the nodes in the network, not on how the information is obtained. So as long as a different connectivity model provides us with enough pairwise distance information for the network, the SDP algorithm should be able to estimate positions with reasonable accuracy. To test this hypothesis, we plan to analyze the results of applying the model on empirical data, that is, actual pairwise distance information obtained from existing networks.

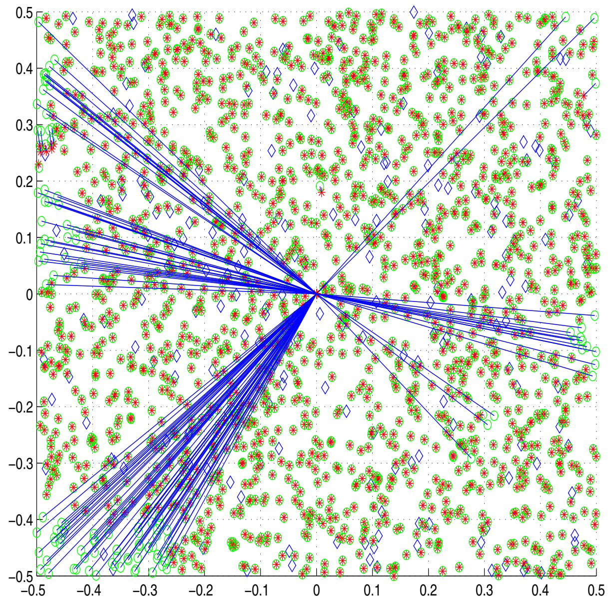
8. REFERENCES

[1] A. Y. Alfakih, A. Khandani, and H. Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. *Comput. Optim. Appl.*, 12(1-3):13–30, November 1999.

[2] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[3] D. Bertsimas and Y. Ye. Semidefinite relaxations, multivariate normal distributions, and order statistics. *Handbook of Combinatorial Optimization*, 3:1–19, 1998.

[Intermediate result: position estimation after the first two iterations.]



[Final result: position estimation after the first four iterations.]

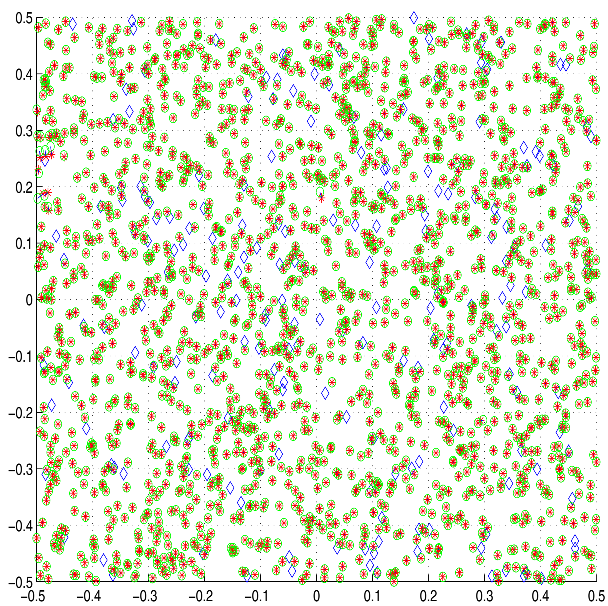


Figure 6: Position estimations of 200 **anchors and** 2000 **sensors, cluster number=64, radio-range=0.06, noisy factor=0.**

[4] P. Biswas and Y. Ye. A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization. Technical report, Dept of Management Science and Engineering, Stanford University, October 2003.

[5] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM., 1994.

[6] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. Technical report, Computer science department, University of Southern California, April 2000.

[7] L. Doherty, L. E. Ghaoui, and S. J. Pister. Convex position estimation in wireless sensor networks. In *IEEE Infocom*, volume 3, pages 1655 –1663, April 2001.

[8] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical report, University of California, Los Angeles, 2002.

[9] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, August 2001.

[10] A. Howard, M. Mataric, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, volume 3, pages 1055–1060, October 2001.

[11] M. Laurent. Matrix completion problems. *The Encyclopedia of Optimization.*, 3:221–229, 2001.

[12] J. Moré and Z. Wu. Global continuation for distance geometry problems. *SIAM Journal on Optimization.*, 7:814–836, 1997.

[13] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *IEEE GLOBECOM (1)*, pages 2926–2931, 2001.

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

[15] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Mobile Computing and Networking*, pages 166–179, 2001.

[16] A. Savvides, H. Park, and M. B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 112–121. ACM Press, 2002.

[17] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM international symposium on Mobile Ad Hoc Networking & Computing*, pages 201–212. ACM Press, 2003.

[18] J. F. Sturm. Let sedumi seduce you, October 2001.

[19] G. Xue and Y. Ye. An efficient algorithm for minimizing a sum of euclidean norms with applications. *SIAM Journal on Optimization.*, 7:1017–1036, 1997.