DISTRIBUTED SENSOR NETWORK LOCALIZATION WITH INACCURATE ANCHOR POSITIONS AND NOISY DISTANCE INFORMATION

Seshan Srirangarajan, Ahmed H. Tewfik and Zhi-Quan Luo

Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455
{seshan, tewfik, luozq@umn.edu}

ABSTRACT

The goal of the sensor network localization problem is to determine positions of all the sensor nodes in a network given certain pairwise *noisy* distance measurements and *inaccurate* anchor node positions. A two-step distributed localization approach based on second-order cone programming (SOCP) relaxation is presented. In the first step, the sensor nodes determine their positions based on local information and in the second step, the anchor nodes refine their positions using information from the neighboring nodes. Our numerical study shows that the sensor and anchor positions cannot be estimated in a single step; the sensors must be estimated first for the results to converge. The second step enables anchors which are in the convex hull of their neighbors to refine their positions. Extensive simulation results with inaccurate anchor positions and noisy distance measurements are presented. These illustrate the robustness of the algorithm and the performance gains achievable in terms of problem size reduction, computational efficiency and localization accuracy.

Index Terms— Distributed algorithms, Relaxation methods, Convex optimization, Positioning, Localization

1. INTRODUCTION

Recent advances in micro-electro-mechanical systems (MEMS) and wireless communication technology has made the large-scale deployment of wireless sensor networks possible. Some of the application areas for sensor networks are industrial automation (process control), military (real-time monitoring of troop movements), utilities (automated meter reading), building control and environmental monitoring.

In most applications, the data reported by the sensors is relevant only if tagged with the accurate location of the sensor nodes. Thus knowledge of the node positions becomes imperative. Research efforts are currently focussed on developing cost-effective techniques to determine node positions using distance measurements between neighboring nodes. This distance information can be obtained via time of arrival, received signal strength or other techniques. We focus on the problem of finding the node positions given this distance information.

The sensor network localization problem can be stated as follows. Assuming knowledge of the positions of some nodes (called anchors) and some pairwise distance measurements, determine the position of all sensor nodes in the network. (Nodes whose positions are unknown will be referred to as sensor nodes). In practice, due to resource constraints on the sensor nodes, the distance measurements are inaccurate or noisy. In addition, the anchor node positions are inaccurate even when determined with the use of GPS or other techniques. A number of methods, based on minimizing some global error function, have been explored to account for the measurement uncertainties. It is observed that the computational complexity varies based on the optimization model chosen. Most approaches in the literature do not account for inaccuracies in the anchor positions.

The localization problem in its original form is a non-convex optimization problem (shown in the next section) and it will be relaxed to a convex problem. Biswas and Ye proposed the semidefinite programming (SDP) relaxation [1]. The SDP relaxation approach can solve small problems effectively. The authors report a few seconds of PC execution time for a 50 node network. They have also proposed two techniques to improve the accuracy of the SDP solution [2]. However, the number of constraints in the SDP model is $O\left(n^2\right)$, where n is the number of nodes in the network

Most SDP solvers can handle problems with at most 100 variables, while sensor networks typically have 100's of nodes resulting in problem dimensions in the 10,000's. To overcome this difficulty, Biswas and Ye proposed a distributed method for solving the SDP [3]. In this iterative distributed scheme, the anchors are first partitioned into many clusters according to their physical locations. A sensor is assigned to a cluster if the sensor has a direct link to one of the anchors. Then semidefinite programs are solved independently for each cluster. The sensors whose position becomes known are used to iteratively decide the remaining un-positioned sensors. The authors report a few minutes of PC execution time for a network with 4000 nodes. But, since the clustering is done based on geographic locations, each cluster may have only partial connection information for the border sensors if these have connections with multiple clusters. Thus border sensors may not get positioned accurately.

We consider the second-order cone programming (SOCP) relaxation due to its simpler structure and the potential to be solved faster. The SOCP relaxation for the localization problem was first studied by Tseng [4]. We propose a technique that enables the SOCP relaxation problem to be solved in a completely *distributed* fashion. In the first step, each sensor node executes the localization algorithm independently using distance information to the anchors and sensors with which it is directly linked (or which are within its communication range). In a second step, the anchors use the position information from the neighboring nodes and the associated distance information to refine their positions. The results show that this second step results in a significant improvement in positioning of the inner anchors. The problem dimension is reduced to a linear function of the number of neighbors of a node, which enables each sensor node to determine its location. The performance gains are achieved without sacrificing localization accuracy.

2. SENSOR NETWORK LOCALIZATION: PROBLEM FORMULATION

The localization problem is mathematically formulated as follows. Consider *n* distinct points in R^d ($d \geq 1$). Given the positions of the last $(n-m)$ points (or anchors) $x_{m+1}, ..., x_n$ and the Euclidean distances d_{ij} between neighboring points i and j where $(i,j) \in A$. A is the neighbor set defined as $A = \{(i,j) :\| x_i - x_j \| \leq RadioRange\}$ [1], we need to estimate the positions of the first m points (sensors). This can be formulated as the following non-convex minimization problem:

$$\min_{x_1,\ldots,x_m} \sum_{(i,j)\in A} \left| \, \|x_i - x_j\|^2 - d_{ij}^2 \right| \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm.

3. SECOND-ORDER CONE PROGRAMMING RELAXATION

The original problem (1) is non-convex but can be reformulated in convex form using relaxation techniques. As a first step, (1) can equivalently be written as:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} |y_{ij} - d_{ij}^2| \;\text{ s.t. } y_{ij} = \|x_i - x_j\|^2, \forall (i,j) \in A$$

Relaxing the equality constraints to "$\geq$" inequality constraints yields the following convex problem:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} |y_{ij} - d_{ij}^2| \;\text{ s.t. } y_{ij} \geq \|x_i - x_j\|^2, \forall (i,j) \in A \tag{2}$$

[1] The set A is undirected: $(i,j) = (j,i), \forall (i,j) \in A$

which is an SOCP. Tseng has shown in [4] that even though the SOCP relaxation is weaker than the SDP relaxation, it can accurately position (up to square distance error) a large percentage of the sensors. The problem in (2) can equivalently be written as:

$$\begin{aligned} \min_{x_1,\ldots,x_m,y_{ij},t_{ij}} & \sum_{(i,j)\in A} t_{ij} \\ \text{s.t.}\, y_{ij} &\geq \|x_i - x_j\|^2 \qquad \forall (i,j) \in A \\ t_{ij} &\geq |y_{ij} - d_{ij}^2| \qquad \forall (i,j) \in A \end{aligned} \tag{3}$$

A distributed localization algorithm is presented next.

4. DISTRIBUTED ALGORITHM AND IMPLEMENTATION

In a distributed algorithm, implemented over multiple processors, the algorithm is divided into "phases". During each phase, every processor must execute a number of computations that depend on the results of the computations of other processors in previous phases. However, the timing of computations at any one processor during a phase can be independent of the timing of the computations at other processors within the same phase. All interactions between processors takes place at the end of the phases. Such distributed algorithms are also called *synchronous*. Here we show how the SOCP relaxation for the sensor network localization problem can be formulated as a synchronous distributed algorithm.

Let $N_A(i) = \{j : (i,j) \in A\}$ be the neighbor set for node x_i. The problem in (3) can be solved independently over the m sensor nodes x_i, where each node uses information (x_j, d_{ij}) from its neighboring nodes x_j, $j \in N_A(i)$. The information exchange between a node and its neighbors takes place at the end of each iteration (or phase). Thus (3) decomposes to the following distributed formulation:

$$\begin{aligned} \min_{x_i,y_{ij},t_{ij}} & \sum_{j\in N_A(i)} t_{ij} \\ \text{s.t.}\, y_{ij} &\geq \|x_i - x_j\|^2 \qquad \forall j \in N_A(i) \\ t_{ij} &\geq |y_{ij} - d_{ij}^2| \qquad \forall j \in N_A(i) \end{aligned} \tag{4}$$

This can be written in standard SOCP form as:

$$\begin{aligned} \min_{x_i,y_{ij},t_{ij}} & \sum_{j\in N_A(i)} t_{ij} \\ \text{s.t.} & \left(\frac{y_{ij} + t_i^{'}}{2}\right)^2 \geq \left(\frac{y_{ij} - t_i^{'}}{2}\right)^2 + \|x_i - x_j\|^2 \\ & t_{ij} \geq |y_{ij} - d_{ij}^2| \qquad \forall j \in N_A(i) \\ & t_i^{'} = 1 \end{aligned} \tag{5}$$

The obvious way to solve the localization problem is to solve (5) simultaneously at each of the sensors and anchors.

We ran a few simulation test cases using this approach, but the results did not converge in each of those cases [2]. Hence we adopt a two-step approach. In the first step, each sensor node estimates its position using distance information from its neighbors and in the second step, the anchor nodes use information from their neighbors to refine their positions.

Let $N_i(=|N_A(i)|)$, represent the cardinality of set $N_A(i)$. The SOCP (5) has $2N_i + 3$ variables, $2N_i$ conic constraints and 1 equality constraint. In sensor networks, due to the relatively short radio range of the sensors, the number of *neighbors* of a given node is a small fraction of the total number of nodes in the network. Thus (5) results in significantly smaller problem sizes than approaches proposed in the literature. The SOCP (5) can be efficiently solved in practice by interior point methods. Here we use SeDuMi [5] to solve this problem.

5. SIMULATION RESULTS

In this section, we present simulation results based on the SOCP relaxation (5). We generate the true positions of the sensors and the anchors $x_1^t, \ldots, x_n^t$ independently according to a uniform distribution on the unit square $[-0.5, 0.5]^2$.

$$A = \{(i,j) : \|x_i^t - x_j^t\| \le RadioRange\}$$

$$d_{ij} = \|x_i^t - x_j^t\| \cdot \max\{0, 1 + \epsilon_{ij} \cdot nf_d\} \quad \forall (i,j) \in A$$

$$x_i = x_i^t \cdot \max\{0, 1 + \epsilon_{ij} \cdot nf_a\} \quad \forall i = (m+1), \ldots, n.$$

where ϵ_{ij} is a random variable representing measurement noise (normally distributed), $RadioRange \in (0,1)$, nf_a and $nf_d \in [0,1]$ are the noise factors for anchor positions and distance measurements, respectively. p is the percentage of anchors.

We wrote the code in Matlab to solve the SOCP relaxation. Our code calls SeDuMi (Version 1.1) [5]. Simulations were carried out on a PC with 3 GHz Pentium 4 processor and 2 GB RAM running Matlab 7.2.0 (R2006a).

Table 1 lists the input parameter values for the different test cases. Also reported are the maximum SOCP (5) dimension and the corresponding CPU time per sensor node (excluding the time for computing the relative distances d_{ij}). Comparing with the SOCP dimensions reported by Tseng in [4] for similar network sizes, the dimensions reported here are smaller by at least two orders of magnitude. The SOCP dimension, which depends on number of neighboring nodes, scales well with the network size. As a result, the cpu time and per node computational burden is significantly reduced.

To check the accuracy of our algorithm, we compute the average positioning error for the sensor and anchor nodes. Figure 1 shows that the error increases slightly for nf_d larger than 0.05. We fixed $nf_d = 0.01$ for the rest of the simulations to understand the effect of the other parameters.

[2]Due to space constraints, we are unable to elaborate on the results showing divergence under simultaneous algorithm execution at the anchors and sensors.

Test Case	n	nf_a	Max. SOCP dimension	CPU time per node (in sec)
1	500	0.10	115 × 169	1.03
2	500	0.20	127 × 187	1.02
3	500	0.30	107 × 157	1.06
4	1000	0.10	175 × 259	1.22
5	1000	0.20	191 × 283	1.24
6	1000	0.30	175 × 259	1.27
7	2000	0.10	339 × 505	1.43
8	2000	0.20	375 × 559	1.56
9	2000	0.30	339 × 505	1.57

Table 1. SOCP relaxation (5) dimensions with corresponding CPU times. ($p = 0.20$, $RadioRange = 0.10$, $nf_d = 0.01$)

The true and the estimated node positions for test case 5 are shown in Figure 2. There is a close match between the estimated and true positions for sensors which lie in the convex hull of their neighbors. The estimated positions become less accurate as we move towards the boundary.

Figure 3 shows the effect of noise factor nf_a and network size (n) on the average error. The average error increases slightly as nf_a increases from 0.10 (i.e., each anchor coordinate has up to $\pm 10\%$ error) to 0.30. As the network size (n) increases resulting in higher node connectivity, the average error decreases. Figure 4 shows that increasing the percentage of anchors lowers the positioning error for both the sensor and anchor nodes. The estimated positions are on average within 5-6% of the true positions over a range of parameter values.

6. FUTURE RESEARCH DIRECTION

The algorithm presented refines the anchor positions in the second step. This improvement is significant only for the in-

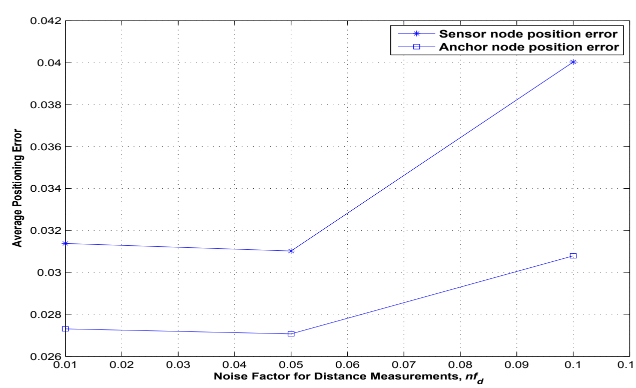

Fig. 1. Average positioning error as a function of the Noise Factor nf_d.($n = 1000$, $RadioRange = 0.10$, $p = 0.20$ and $nf_a = 0.10$)

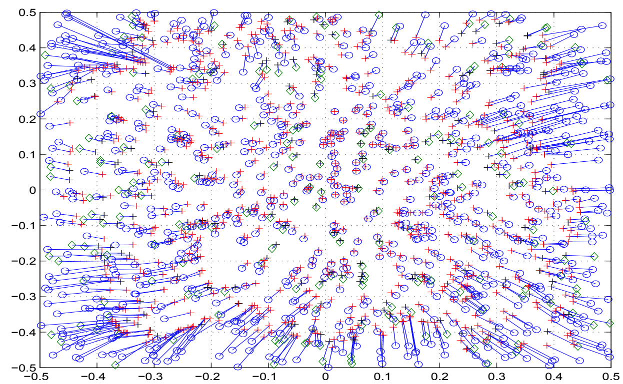

Fig. 2. Results for Test Case 5. True positions of sensors (∘) and anchors (⋄) with the estimated positions (+). Solid lines indicate error between the estimated and true node positions.

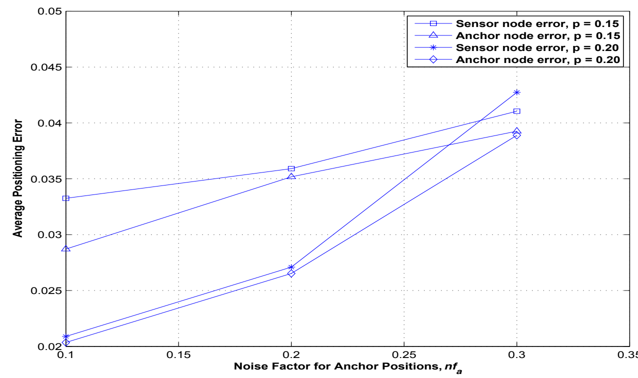

Fig. 4. Average positioning error as a function of nf_a and p. ($n = 2000$ and $nf_d = 0.01$)

ner anchors. The simulation results presented here are conservative, since we have assumed the same radio range for the sensor and anchor nodes. In reality, anchor nodes tend to have more transmit power (and a larger range). Our preliminary results suggest that having a separate phase where the anchors talk to each other gives improved positioning of the inner anchors. An analytical study of the proposed distributed algorithm will appear in a future IEEE publication.

7. CONCLUSION

The proposed distributed SOCP approach solves the localization problem, in the presence of inaccuracies in anchor position and distance measurements, with great accuracy and speed for very large problem sizes. This method also improves the positioning of the anchors which are in the convex hull of their neighbors.

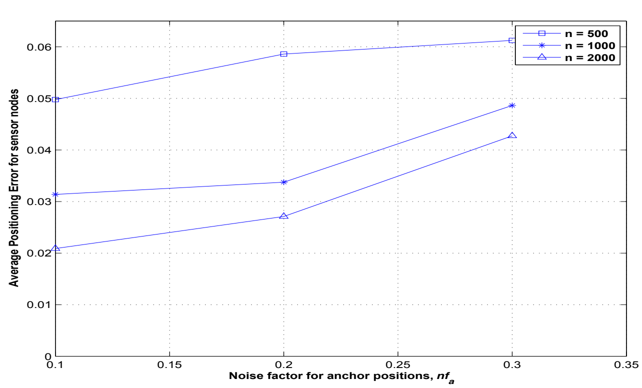

Fig. 3. Average positioning error as a function of nf_a and n. ($p = 0.20$, $RadioRange = 0.10$, $nf_d = 0.01$)

8. REFERENCES

[1] P. Biswas and Y. Ye. Semidefinite Programming for Ad Hoc Sensor Network Localization, IPSN, April 2004.

[2] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang and Y. Ye. Semidefinite Programming Approaches for Sensor Network Localization with Noisy Distance Measurements, to appear in IEEE Trans. Automation Science and Engg.

[3] P. Biswas and Y. Ye. A Distributed Method for Solving Semidefinite Programs Arising from Ad Hoc Wireless Sensor Network Localization, in Multiscale Optimization Methods and Applications, Springer Verlag, Jan 2006.

[4] P. Tseng. Second-Order Cone Programming Relaxation of Sensor Network Localization, submitted to SIAM Journal on Optimization.

[5] J. Strum. Using SeDuMi 1.02, A Matlab Toolbox for Optimization Over Symmetric Cones, Oct 2001.

[6] L. Doherty, K. S. J. Pister and L. E. Ghaoui. Convex Position Estimation in Wireless Sensor Networks, IEEE INFOCOM 2001, April 2001.

[7] M. W. Carter, H. H. Jin, M. A. Saunders and Y. Ye. SPASELOC: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization, submitted to SIAM Journal on Optimization.

[8] P. Biswas, T.-C. Lian, T.-C. Wang and Y. Ye. Semidefinite Programming Based Algorithms for Sensor Network Localization, ACM Trans. on Sensor Networks, May 2006.

[9] S. Boyd and L. Vandenberghe. Convex Optimization, Cambridge University Press, 2004.

[10] D.P. Bertsekas and J.N. Tsitsiklis. Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, 1989.