Distributed Sensor Network Localization Using SOCP Relaxation

Seshan Srirangarajan, *Member, IEEE,* Ahmed H. Tewfik, *Fellow, IEEE,* and Zhi-Quan Luo, *Fellow, IEEE*

***Abstract*—The goal of the sensor network localization problem is to determine positions of all sensor nodes in a network given certain pairwise *noisy* distance measurements and some anchor node positions. This paper describes a distributed localization algorithm based on second-order cone programming relaxation. We show that the sensor nodes can estimate their positions based on local information. Unlike previous approaches, we also consider the effect of inaccurate anchor positions. In the presence of anchor position errors, the localization is performed in three steps. First, the sensor nodes estimate their positions using information from their neighbors. In the second step, the anchors refine their positions using relative distance information exchanged with their neighbors and finally, the sensors refine their position estimates. We demonstrate the convergence of the algorithm numerically. Simulation study, for both uniform and irregular network topologies, illustrates the robustness of the algorithm to anchor position and distance estimation errors, and the performance gains achievable in terms of localization accuracy, problem size reduction and computational efficiency.**

***Index Terms*—Distributed algorithms, convex optimization, relaxation methods, second-order cone programming, positioning, localization, synchronous and asynchronous algorithms.**

I. Introduction

RECENT advances in micro-electro-mechanical systems (MEMS) and wireless communication technology has made possible the large-scale deployment of wireless sensor networks with thousands of nodes. 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 [1].

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. Using nodes equipped with Global Positioning System (GPS) is a costly option. This research focusses 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 [2], [3].

Manuscript received February 27, 2007; revised August 15, 2007 and February 14, 2008; accepted March 1, 2008. The associate editor coordinating the review of this paper and approving it for publication was R. Fantacci. This research is supported in part by the National Science Foundation, grant no. DMS-0610037 and in part by the USDOD ARMY, grant no. W911NF-05-1-0567.

S. Srirangarajan is with the Intelligent Systems Center, Nanyang Technological University (e-mail: seshan@ntu.edu.sg).

A. H. Tewfik and Z.-Q. Luo are with the Department of Electrical and Computer Engineering, University of Minnesota.

Digital Object Identifier 10.1109/T-WC.2008.070241

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. We will refer to nodes whose positions are unknown as the 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 may be inaccurate even when determined with the use of GPS or other techniques. Most approaches in the literature do not account for inaccuracies in the anchor positions. 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.

In this paper, we present a distributed algorithm based on second-order cone programming (SOCP) for solving the sensor network localization problem. In the presence of distance estimation errors each sensor node determines its position by executing the localization algorithm independently using distance information to the anchor and sensor nodes with which it is directly linked (i.e., which are within its communication range). If in addition to the distance estimation errors, the anchor positions also have errors then the algorithm consists of three steps: using the local distance information and inaccurate anchor positions each sensor node estimates its position. Then, the anchors execute the localization algorithm using position information from their neighboring nodes and the associated distance information to refine their positions. Finally, the sensor nodes execute the localization algorithm to refine their position estimates [4].

One of the significant advantages of our approach is that it is fully distributed and converges to an optimal (or near-optimal) solution. As a result of the distributed nature of the solution, the problem dimension at each node is a linear function of only the number of neighbors of the node. There is no significant increase in the computational effort per node even in large networks (for a given node connectivity level), whereas most existing methods result in an exponential increase in the computation time with network size. Thus, the distributed SOCP approach is suitable for large-scale networks with thousands of nodes. As we will demonstrate, the performance gains are achieved without sacrificing localization accuracy. We demonstrate the convergence of our algorithm numerically.

The rest of the paper is organized as follows. Section II provides an overview of existing approaches. Section III presents the mathematical formulation and the SOCP relaxation of the localization problem. In section IV we present

the distributed localization algorithm based on the SOCP relaxation. The simulation study appears as sections V and VI. Section VII explores the problem of tracking a mobile node in indoor environments. Finally, in section VIII we discuss the asynchronous version of the distributed algorithm.

II. Related Work

Sensor network localization has been an area of active research in recent years with a large number applications. Some survey articles on this research area are [2], [3], [5], [1]. Most localization systems estimate the node positions using some kind of range or distance information between nodes. However, some systems such as [5] perform localization using connectivity information. Such systems rely on high anchor density and result in relatively low positioning accuracy.

We will show in the next section that the localization problem in its original form is a non-convex optimization problem with many local minima. Doherty et al. [6] formulate the localization problem as a feasibility problem with convex radial constraints. However, this method requires centralized computation which is not suitable for large networks. Shang proposed a distributed localization method MDS-MAP(P, R) based on multi-dimensional scaling (MDS) [7]. This method builds for each node a local map of the small subnetwork in the node's vicinity and then merges these local maps to form a global map followed by a refinement step. This method needs only a few anchors (generates a relative map in the absence of anchors) and partly overcomes the drawback associated with centralized computation. However, the construction of local maps for each node results in enormous amount of redundant computation as most local maps are not used in building the global map. It is not entirely suitable for large networks as the cost of refining and merging the local maps grows faster than linear due to the larger maps being manipulated. Also the cost of refining the global map (optional), a single global optimization step, becomes dominant for large networks.

Costa et al. [8] apply distributed weighted MDS (dwMDS) to the localization problem and formulate the problem using a general form of the cost function we use in this paper. They solve the minimization problem using majorizing functions.

Biswas and Ye solve the problem using the semidefinite programming (SDP) relaxation [9]. This 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 [10]. The first technique adds a regularization term to the objective function to force the SDP solution to lie close to a low dimensional subspace of R^d and the second technique improves the SDP estimated solution using a gradient-descent method. 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 [11]. 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 sensors with unknown position. 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 [12], each cluster may have only partial connection information for the border sensors if these have connections with multiple clusters. Thus sensors on the border of each cluster may not get positioned accurately [13].

We consider the second-order cone programming (SOCP) relaxation due to its simpler structure and the potential to be solved faster. SOCP relaxation for the localization problem was first studied by Tseng [14]. There it was shown that even though the SOCP relaxation is weaker than the SDP relaxation, it can accurately position a large percentage of the sensors. The localization approach presented in this paper enables the SOCP relaxation problem to be solved in a completely *distributed* fashion. Each sensor node executes the localization algorithm independently using distance information to the anchors and sensors with which it is directly linked.

A number of existing approaches consider the distance estimation errors [15], [16], however most do not consider the inaccuracy in anchor positions which is also a significant source of error. Anchors are typically positioned using GPS or by means of surveying by humans. Civilian GPS accuracy is limited to about 15m while surveying is prone to human observation errors. In section VI, we demonstrate that the distributed SOCP approach provides good localization accuracy even in the presence of significant error in the anchors positions.

III. Sensor Network Localization: Problem Formulation

The localization problem is mathematically formulated as: Consider n distinct nodes in R^d ($d \geq 1$). Given the positions of the last $(n-m)$ nodes (anchors) $x_{m+1}, ..., x_n$ and the Euclidean distance d_{ij} between neighboring nodes i and j where $(i,j) \in A$. A is the undirected neighbor set defined as $A := \{(i,j) : \|x_i - x_j\| \leq RadioRange\}$. We need to estimate the positions of the first m nodes (sensors). This can be formulated as the following non-convex minimization [17]:

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

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

(1) 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 "greater than or equal to" inequality constraints yields the following convex problem:

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

which is an SOCP. (2) can equivalently be written as:

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

One approach would be to solve the SOCP problem as a global minimization over the entire network. However, due to the problem sizes encountered in sensor networks solving the SOCP relaxation *globally* might be computationally demanding. In addition, a parallel or distributed algorithm is always preferred in sensor networks, which we propose next.

IV. Distributed SOCP Localization Algorithm

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 take 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 [18].

We can approximately reformulate (2) as:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} \{|\, y_{ij} - d_{ij}^2 \,| + I_-(f_{ij}(x,y))\}$$

$$\text{where } f_{ij}(x,y) = \|x_i - x_j\|^2 - y_{ij} \text{ and}$$

$$I_-(u) = \begin{cases} 0 & u \leq 0, \\ \infty & u \geq 0. \end{cases}$$

I_- is the indicator function which can be approximated by the *logarithmic barrier* function and the problem reduces to:

$$\min_{x_1,\ldots,x_m,y_{ij}} \sum_{(i,j)\in A} \{|\, y_{ij} - d_{ij}^2 \,| - (1/t)\log(y_{ij} - \|x_i - x_j\|^2)\}. \tag{4}$$

It is seen that the objective function in (4) is separable. For each $i \in \{1, ..., m\}$, the objective function depends only on the positions of the neighboring nodes and the pairwise distance measurements between them. This enables the objective function to be decomposed and the minimization can then be carried out at each sensor node x_i using local information. Each sensor can independently solve this minimization using information from its neighboring sensor nodes and anchors.

Let $N_A(i) = \{j : (i,j) \in A\}$ be the neighbor set for node x_i. Using the separability observation, (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 nodes occurs at the end of each iteration (or phase). Thus (3) decomposes to the following distributed formulation:

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

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 distributed SOCP algorithm consists of a phase where each sensor node estimates its position using local information and solving the SOCP (5). In an iterative distributed scheme, this would be followed by a communication phase wherein each node exchanges its position estimate with its neighbors. These iterations are repeated after fixed intervals of time or when any new information becomes available at a node. It should be noted that the algorithm uses information from neighboring anchors as well as sensors to position a given node. Thus to obtain a non-trivial position estimate each node needs at least 3 neighbors (for 2-D localization) with position estimates, as opposed to the more stringent requirement of having 3 anchors in the neighborhood that many triangulation/trilateration schemes impose.

If the anchor positions are inaccurate, the distributed SOCP approach will consist of three steps: after the sensor nodes estimate their positions based on the inaccurate anchor positions and distance information, the anchors solve the local SOCP (5) using position information from their neighboring nodes and the associated distance information to refine their positions. As we will show, this second step results in a significant improvement in the positioning accuracy of the inner anchors. Finally, another iteration of the local SOCP (5) over the sensor nodes further refines their position estimates.

Let $n_i (= |N_A(i)|)$ represent the number of neighbors of the node x_i. SOCP (5) has $2n_i + 3$ variables, $2n_i$ conic constraints and 1 equality constraint. In sensor networks, due to the 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 (i.e., $n_i \ll n$). Thus, the distributed SOCP approach results in significantly smaller problem sizes than approaches proposed in the literature. The SOCP problem (5) can be efficiently solved by interior point methods. Here we use SeDuMi [19] to solve this problem. The details on how to rewrite SOCP (5) in SeDuMi form are given in Appendix.

V. Localization with Accurate Anchor Position Information

The experiments in this section assume that accurate anchor position information is available. In this setting, the localization problem is solved by executing the distributed SOCP algorithm at each of the sensor nodes. We assess the average-case performance on networks with uniform topology as well as irregular topology. For each parameter setting, the algorithm is run on 5 randomly generated examples.

We randomly generate the true positions of the sensors and anchors $x_1^t, \ldots, x_n^t$ according to a uniform distribution on the unit square $[-0.5, 0.5]^2$ and noisy distance measurements d_{ij}

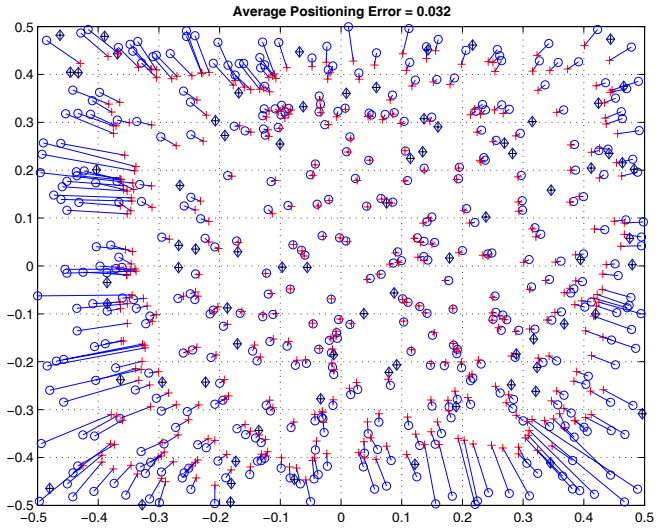


Fig. 1. Distributed SOCP results for Uniform topology: $n = 500$, $RadioRange = 0.15$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.032$ and $err_{max} = 0.232$.

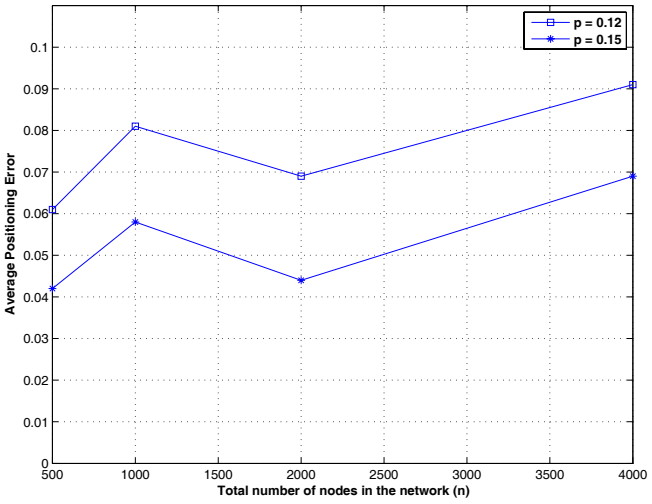


Fig. 2. Average positioning error as a function of the network size (n) for two different percentages of anchors. ($RadioRange = 0.10$, $nf_d = 0.05$ and average node connectivity ≈ 30)

are generated by adding normally distributed measurement noise to the true distance. Specifically:

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

where ϵ_{ij} is a normal random variable $\mathcal{N}(0,1)$ representing measurement noise, $RadioRange \in (0,1)$ represents the radio range of the nodes and $nf_d \in [0,1]$ is the noise factor (standard deviation of the distance error in percentage) for the distance measurements. For a standard deviation of 10% in the distance estimation error we set $nf_d = 0.10$.

We wrote the code in Matlab to solve the SOCP relaxation. Our code calls SeDuMi (Version 1.1) [19], a C implementation of a predictor-corrector primal-dual interior point method for solving SDP/SOCP. The simulations were carried out on a PC with 2.53 GHz Pentium 4 processor and 1 GB RAM running Matlab 7.0.1 (R14).

To check the positioning accuracy of our algorithm, we compute the average (sensor) positioning error ($\overline{err}$). First we consider the uniform topology. Fig. 1 shows the results for a randomly generated 500 node network with $nf_d = 0.05$. True positions of the sensors and anchors are depicted using $\circ$ and $\diamond$, respectively. The estimated node positions are depicted with +. Solid lines indicate the error between the estimated and true sensor positions. A close match is observed between the estimated and true positions for sensors which lie in the convex hull of their neighbors. The average positioning error is 0.032 (21.3% of $RadioRange$). The estimated positions become less accurate as we move towards the boundary.

Fig. 2 shows the effect of the percentage of anchors (p) on the average positioning error for different network sizes with similar node connectivity levels. Increasing p from 12% to 15% lowers the average error. The variation in the average error across network sizes (for a given p) is most likely due to the small differences in the node connectivity level.

The positioning accuracy improves significantly with increase in the node connectivity. Fig. 3 shows that the average

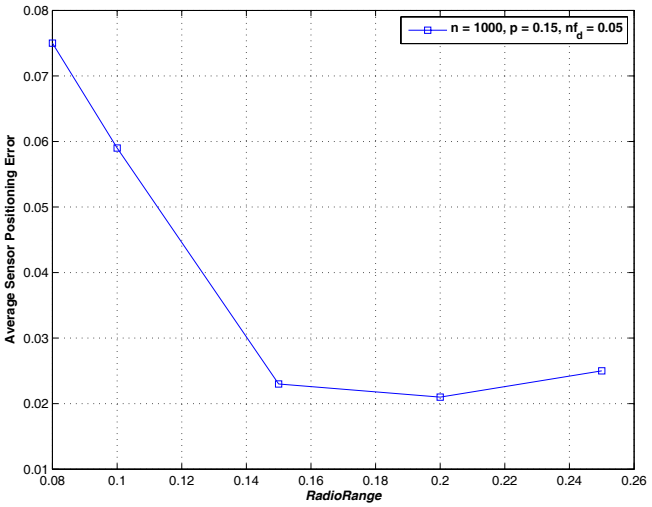


Fig. 3. Average positioning error as a function of $RadioRange$. ($n = 1000$, $p = 0.15$ and $nf_d = 0.05$)

error decreases steadily as $RadioRange$ is increased from 0.08 to 0.20. For larger radio ranges the error tends to reach a lower bound determined by the distance estimation errors.

Irregular topologies are more difficult than uniform topologies. The results for an irregular topology, namely a C-shaped network, with 300 nodes is shown in Fig. 4. The average positioning error is 0.048 (32% of the $RadioRange$).

Table I lists the test cases used to understand the computational complexity of the distributed SOCP algorithm. $RadioRange$ is chosen such that the average node connectivity is about the same across all test cases. Table I lists the various parameter settings, the average node connectivity, the typical SOCP (5) dimension and the computation time per sensor node (excluding the time needed for computing the relative distances d_{ij} and communication or message exchanges). Comparison with the SOCP dimensions reported by Tseng [14] for similar network sizes reveals that the

TABLE I
DISTRIBUTED SOCP: INPUT PARAMETERS FOR THE TEST CASES, CORRESPONDING SOCP (5) DIMENSIONS AND CPU TIMES. (p GIVES THE PERCENTAGE OF ANCHOR NODES, AND NOISE FIGURE $nf_d = 0.05$ FOR ALL TEST CASES).

Test case	n	$RadioRange$	p	$\|A\|$	Avg. node connectivity	Typical SOCP dimension	CPU time per node (in sec)
1	500	0.15	0.12	15104	30.2	123×181	1.24
2	500	0.15	0.15	15054	30.2	181×124	1.19
3	1000	0.10	0.12	28368	28.4	115×169	1.20
4	1000	0.10	0.15	28158	28.2	115×169	1.18
5	2000	0.08	0.12	73566	36.8	151×223	1.30
6	2000	0.08	0.15	76706	38.4	155×229	1.29
7	4000	0.05	0.15	119082	29.8	123×181	1.03

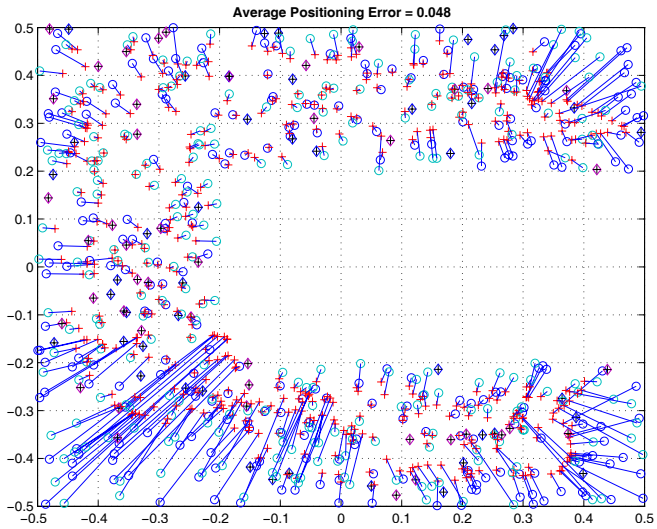


Fig. 4. Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRange = 0.15$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.048$ and $err_{max} = 0.448$.

dimensions in Table I are smaller by at least two orders of magnitude and are still in the realm of problem sizes which can be handled efficiently, even for a network with 4000 nodes. The SOCP dimension, which depends on the number of neighboring nodes, does not increase with the network size for a given node connectivity. Thus, the per node computational burden is significantly reduced.

Table II presents test cases for the MDS-MAP(P, R) algorithm [7]. The cost of refining and merging the local maps, and the optional global refinement step in the MDS-based algorithm becomes dominant for large networks ($n > 300$). Hence, network sizes for these tests are smaller than those for the SOCP algorithm. However, $RadioRange$ is chosen such that the node connectivity is approximately the same as for the test cases in Table I.

From Tables I and II, it is seen that the MDS-MAP(P, R) algorithm requires slightly more than three times the computational effort needed for the distributed SOCP algorithm, even for relatively small network sizes. Thus the distributed SOCP approach significantly improves the computational efficiency without sacrificing localization accuracy.

VI. LOCALIZATION WITH ANCHOR POSITION ERRORS

The experiments in this section consider the inaccuracy in the anchor positions in addition to the distance estimation

TABLE II
MDS-MAP(P, R): INPUT PARAMETERS AND CPU TIMES. (p GIVES THE PERCENTAGE OF ANCHOR NODES, AND $nf_d = 0.05$ FOR ALL TEST CASES).

Test case	n	$RadioRange$	p	Avg. node connectivity	CPU time per node (in sec)
1	100	0.35	0.050	29.6	4.03
2	200	0.25	0.025	32.9	4.57
3	300	0.20	0.017	30.9	4.35
4	500	0.15	0.010	30.8	4.07

errors. The goal is to localize the sensors while reducing the impact of the anchor position errors on the positioning accuracy. One way to achieve this is to solve SOCP (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. Hence we propose a three-step distributed approach. In the first step, each sensor node estimates its position using distance information from its neighbors. In the second step, the anchors use information from their neighbors to refine their positions. We also observed that this second step aids in refining only those anchor positions which are within the convex hull of their neighbors. Thus the anchor refinement step is applied only for anchors in the interior of the network. Finally, we repeat one more iteration of the distributed SOCP algorithm over the sensors using the refined anchor positions. It is seen that the refined anchor positions improve the sensor positioning significantly.

The simulation setting is similar to that in section V except for the following differences. The noisy distance measurements and inaccurate anchor positions are generated as:

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

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

where nf_a and $nf_d \in [0, 1]$ are the noise factors for anchor positions and distance measurements, respectively. Simulations in this section were carried out on a PC with 3 GHz Pentium 4 processor and 2 GB RAM running Matlab 7.2.0 (R2006a).

We experimented with the noise factors (nf_d and nf_a) to understand their effect on the positioning accuracy. Fig. 5 shows variation of the average error with increasing nf_d. The distributed SOCP algorithm handles distance errors as large as 20% gracefully with a small degradation in positioning accuracy. MDS-MAP(P, R) algorithm gives better accuracy using the $O(n^3)$ global refinement step and shows similar

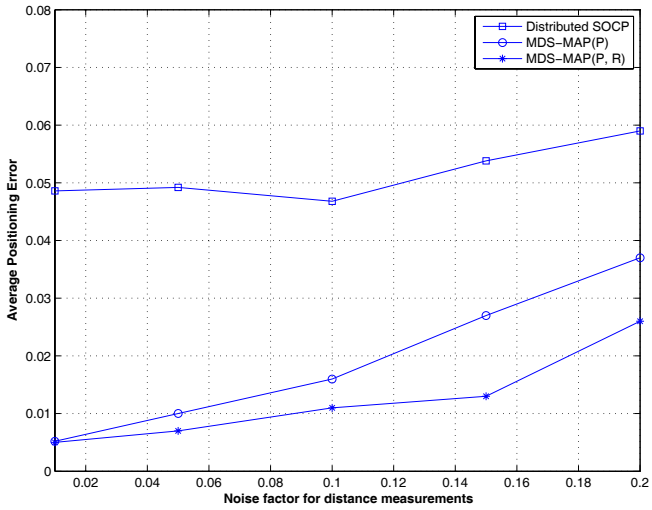


Fig. 5. Average positioning error as a function of the Noise Factor nf_d. ($n = 500$, $RadioRange = 0.15$, $p = 0.15$ and $nf_a = 0.10$)

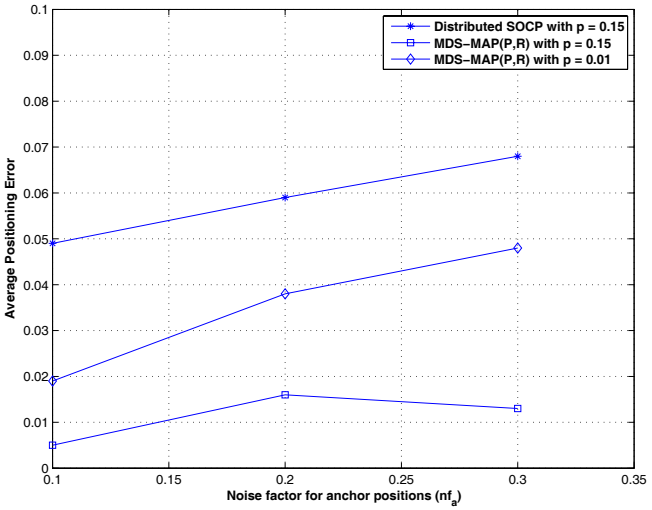


Fig. 6. Average positioning error as a function of nf_a and p. ($n = 500$, $RadioRange = 0.10$ and $nf_d = 0.01$)

degradation. It should be noted that MDS-MAP(P), the MDS-MAP algorithm without global refinement, shows loss in performance compared to MDS-MAP(P, R). MDS-MAP(P) also uses more computations than the distributed SOCP due to the refinement and merging of local maps. It is observed that the singular value decomposition (SVD) step of the MDS-MAP algorithms takes progressively longer to converge as the distance errors increase. Localization systems based on received signal strength (RSS) measurements regularly encounter distance estimation errors of $15-20\%$. The distributed SOCP approach is thus robust to large distance errors while being computationally efficient.

Fig. 6 shows the effect of the noise factor nf_a on the average error. The SOCP algorithm is robust to anchor position errors. The degradation in the positioning accuracy is not significant even in the presence of 30% error in the anchor positions. MDS-MAP(P, R) algorithm which typically works with very few anchors, needs more anchors to compensate for the errors introduced by the inaccurate anchor positions.

In section V we showed that the positioning accuracy can be improved by increasing the radio range of all nodes. Since increasing the radio range of the sensors is typically not feasible due to the resource constraints, increasing the radio range for the anchors is more reasonable. Also, in practice anchors tend to have more transmit power. Fig. 7 illustrates the effect of increasing the radio range of the anchors while keeping the range of the sensor nodes fixed. Increasing the radio range of anchors from 0.10 to 0.15 results in a very significant improvement in the positioning accuracy, but further increase in the anchor radio range does not improve the positioning. This behavior can be explained as follows. During position estimation for a node, we use information from only those sensors and anchors which are within its radio range. We do not account for the fact that some anchors are able to transmit to a sensor node even though the sensor cannot transmit to them (due to asymmetric radio ranges). In effect, we use information from only those nodes which have a bidirectional link with the sensor. So the increased range of

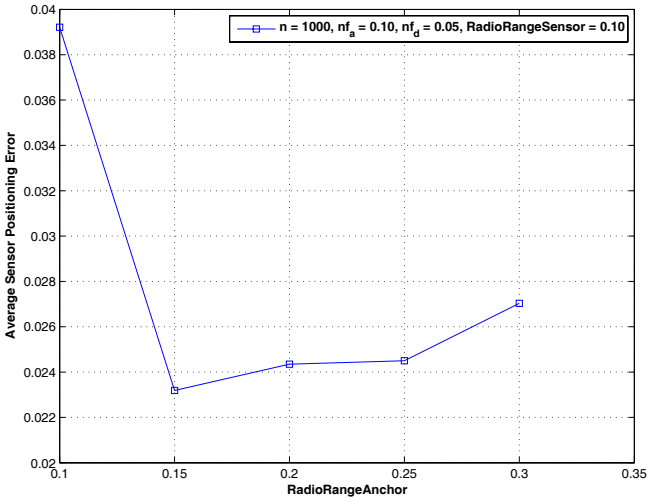


Fig. 7. Average sensor positioning error as a function of the radio range of the anchors ($RadioRangeAnchor$). ($n = 1000$, $p = 0.15$, $nf_d = 0.05, nf_a = 0.10, RadioRangeSensor = 0.10$)

the anchors only aids in refining the position estimates of the anchors when they communicate with other anchors, which improves the sensor positioning but with diminishing return. Further improvement in positioning accuracy can be expected by using information from all anchors which can transmit to a sensor node.

We now revisit the irregular C-shaped network and use longer ranges for the anchors than the sensors. Fig. 8 shows the positioning result when the radio range for the sensors is 0.15 and 0.20 for the anchors. The average positioning error is 0.031 (20.6% of the sensor range), a 10% improvement over the accuracy achieved with the radio range set to 0.15 for both sensors and anchors. Fig. 9 shows the results with the anchor range increased to 0.25 resulting in an average positioning error of 0.025 (16.7% of the sensor range).

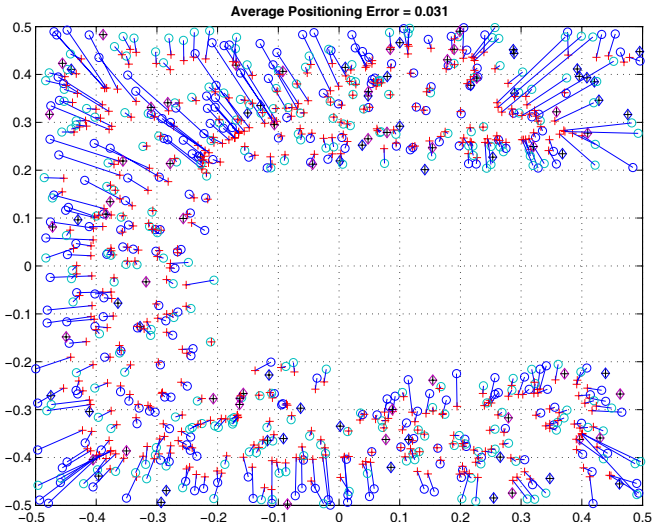


Fig. 8. Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRangeSensor = 0.15$, $RadioRangeAnchor = 0.20$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.031$ and $err_{max} = 0.444$.

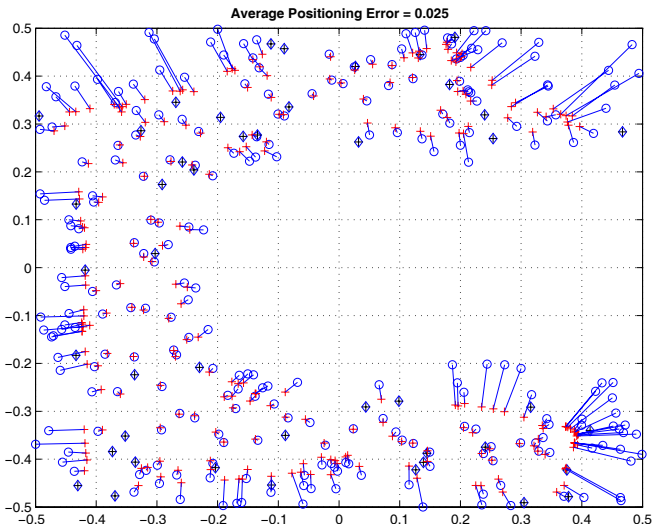


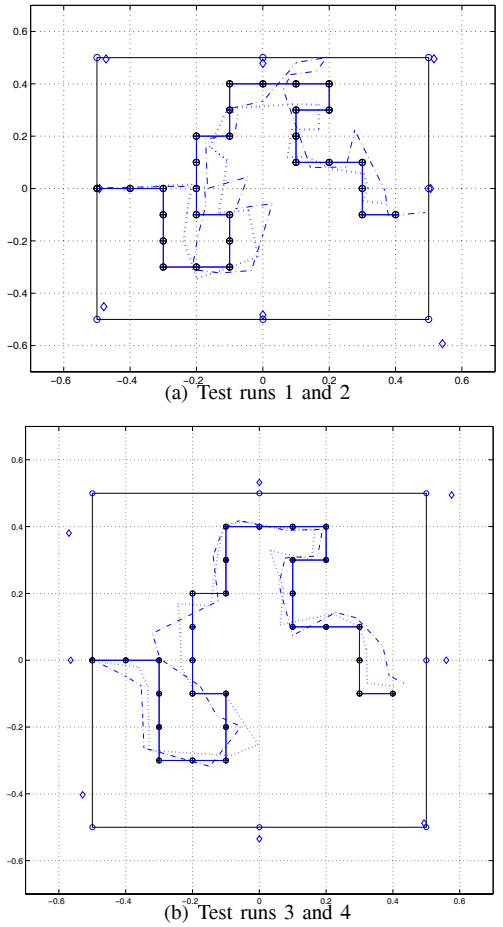
Fig. 9. Distributed SOCP results for irregular (C-shaped) topology: $n = 300$, $RadioRangeSensor = 0.15$, $RadioRangeAnchor = 0.25$, $p = 0.15$ and $nf_d = 0.05$. $\overline{err} = 0.025$ and $err_{max} = 0.429$.

VII. Tracking a Mobile Sensor Node

In this section we consider the problem of tracking a mobile node in an indoor environment using a few high-power and long range anchors. This arises in situations such as tracking fire-fighters who are on a rescue mission inside a building on fire. The fire-fighters can be tracked using wearable sensors and high-power transmitters placed outside the building.

For these experiments, we include a fading coefficient (f) which represents the percentage of total anchors that cannot be heard by the sensor at any given time. This models the obstructions encountered in indoor environments which limit the number of anchors that can be heard at any point.

We use 8 anchor nodes, placed equidistantly at the boundary of a $[-0.5, 0.5]^2$ square grid, for the experiments. The tracking results using the distributed SOCP algorithm are shown in Fig. 10. In Fig. 10(a), the distance estimates have $\pm 15\%$ error standard deviation ($nf_d = 0.15$) whereas in Fig. 10(b), $nf_d = 0.10$. There is no significant degradation in the results as the anchor position error standard deviation is increased from 5% to 15% ($nf_a = 0.05$ to 0.15). Thus, nf_a is fixed at 0.10 for the tracking results shown here. The estimated tracks are fairly accurate with up to 15% error standard deviation in the distance estimates. Distance estimates with more than 20% error standard deviation begin to degrade the results. The error metrics for these experiments are shown in Table III.



(a) Test runs 1 and 2

(b) Test runs 3 and 4

Fig. 10. Tracking Results: Circles ($\circ$) on the $[-0.5, 0.5]^2$ square grid represent the true anchor positions, the diamonds ($\diamond$) represent the inaccurate anchor positions used for the experiments. $\oplus$ indicates sensor node positions along the actual path. Solid lines indicate the actual path followed by the sensor node. Estimated paths are indicated by dash-dot lines for test runs 1, 3 and dotted lines for test runs 2, 4.

TABLE III
SIMULATION PARAMETERS FOR THE TRACKING RESULTS. f IS THE FADING COEFFICIENT, TOTAL NUMBER OF ANCHORS = 8 AND $nf_a = 0.10$.

Test run	nf_d	f	Number of anchors heard	Avg. error	Error std. dev.	Max. error
1	0.15	0.50	4	0.094	0.043	0.182
2	0.15	0.40	5	0.075	0.026	0.152
3	0.10	0.50	4	0.061	0.038	0.169
4	0.10	0.40	5	0.044	0.029	0.117

VIII. Asynchronous Distributed Algorithm

In this section we consider asynchronous execution of the distributed SOCP algorithm. The motivation for this is

TABLE IV
COMPARISON OF SYNCHRONOUS AND ASYNCHRONOUS ALGORITHM EXECUTION. $n = 1000$, $nf_d = 0.05$, $nf_a = 0.10$, $p = 0.15$, $RadioRangeAnchor = 0.15$, $RadioRangeSensor = 0.10$.

	Total number of iterations	CPU time per node (in sec)	Avg. sensor position error	
			Before anchor refinement	After anchor refinement
Sync.	20	1.91	0.0436	0.0232
Async.	60	2.67	0.0447	0.0213

to understand the convergence properties of the algorithm under asynchronous execution and to compare the time to convergence and communication penalty with its synchronous counterpart. We also demonstrate a variation of the asynchronous algorithm that allows the distributed SOCP approach to be used with fewer anchors.

A. Synchronous vs. Asynchronous

Asynchronous algorithms lack the notion of phases and coordination between the different processors is less strict. For asynchronous execution of the distributed SOCP algorithm, we randomly pick the nodes which will update their positions during any given iteration. Each node localizes itself based on whatever information happens to be available from its neighbors at that time; including some information which may not have been updated for the last few iterations.

Table IV presents a representative comparison between the synchronous and asynchronous executions of the algorithm. The asynchronous algorithm needs about three times as many iterations to achieve the same positioning accuracy as the synchronous version. Despite this fact, the computational time per node (excluding the time for communication) increases by a much smaller factor. The communication requirements for the asynchronous version might exceed those for the synchronous execution due to the larger number of iterations resulting in more message exchanges between nodes. This assumes there are no queuing delays affecting the computation. Considering the results in Table IV it can be said that the asynchronous algorithm converges at about the same rate as the synchronous algorithm. This agrees with the analysis of other well-known algorithms such as the Bellman-Ford algorithm.

B. Asynchronous distributed localization with fewer anchors

A distributed localization algorithm offers the possibility of using sensor nodes positioned in one iteration as *pseudo* anchors in the following iterations. The SOCP algorithm makes use of this implicitly. Here we outline a procedure to make explicit use of this, thus allowing the algorithm to localize the nodes using fewer anchors. In a variation of the asynchronous algorithm, instead of randomly choosing nodes which will update their positions, the position update will be based on the availability of at least three neighboring nodes which have obtained an estimate of their position.

Fig. 11 shows the positioning results for a network with $n = 500$ nodes and 5% anchors (or 25 anchors). The anchors are placed on a uniform grid to ensure good coverage. It is seen that the results converge giving an average positioning error of 0.0346 (or 21.6% of the sensor range). This illustrates

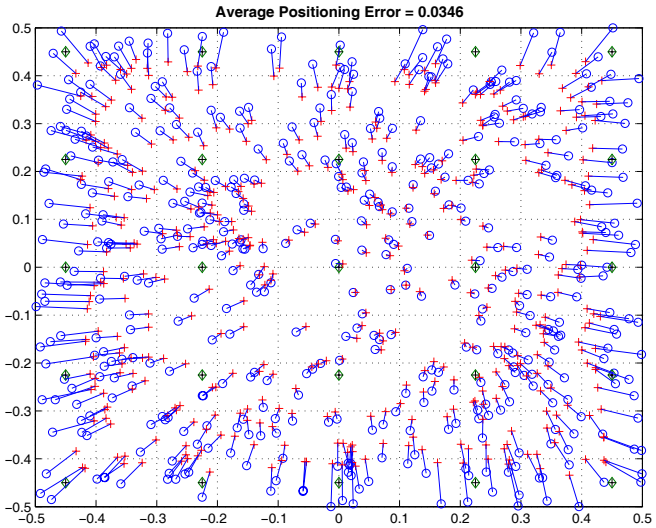


Fig. 11. Asynchronous distributed SOCP using low percentage of anchors: $n = 500$, $p = 0.05$, $RadioRangeAnchor = 0.16$, $RadioRangeSensor = 0.16$, $nf_d = 0.10$ and $nf_a = 0$. $\overline{err} = 0.0346$.

the ability of the SOCP approach to localize nodes with fewer anchors, without loss of positioning accuracy.

Finally, in Table V we provide a comparison of the distributed SOCP, dwMDS [8] and MDS-MAP algorithms [7], on networks with a low percentage of anchors, in terms of performance and complexity. Some of the parameter values are estimated based on the data presented in [7], [8]. dwMDS and distributed SOCP are similar in computational complexity with the SOCP method providing better accuracy. The results using MDS-MAP algorithms, despite using $O(n^3)$ operations, do not differ significantly from the distributed SOCP.

IX. CONCLUSION

The proposed distributed algorithm based on SOCP relaxation solves the localization problem, in the presence of inaccuracies in anchor positions and distance measurements, with significant computational savings and without sacrificing positioning accuracy. An extensive numerical study of the algorithm under different scenarios has been presented. This method is also able to improve positioning of the anchors which are in the convex hull of their neighbors. The asynchronous version of the algorithm also shows good convergence properties and allows localization with fewer anchors.

APPENDIX
SOCP PROBLEM FORMULATION IN SEDUMI FORM

SeDuMi solves problems of the form:

$$\max\ b^T y \quad \text{s.t.} \quad c - A^T y \in K^* \tag{6}$$

where K^* is the dual cone. We now express (5) in this form. Assuming 2-D localization ($d = 2$), we define:

$$L_i := \begin{bmatrix} 1 & 0^{1\times(4n_i+2)} \end{bmatrix}$$

$$r_j := \begin{bmatrix} 0 & d_{ij}^2 \end{bmatrix}^T, t_j := \begin{bmatrix} 0 & 0 & x_{j1} & x_{j2} \end{bmatrix}^T$$

$$S_j := \begin{bmatrix} 0^{1\times(2n_i+3)} & 1_{ij} & 0^{1\times n_i} \\ 0^{1\times(2n_i+3)} & 0^{1\times n_i} & 1_{ij} \end{bmatrix}$$

TABLE V
COMPARISON OF DIFFERENT LOCALIZATION ALGORITHMS ON RANDOM UNIFORM NETWORKS. L IS THE NUMBER OF ITERATIONS NEEDED FOR THE ALGORITHMS TO CONVERGE (TYPICALLY A SMALL NUMBER). † EXPERIMENTAL DATA AND RESULTS FROM [8]. ‡ SIMULATION RESULTS FROM [7].

	n	nf_d	p	$RadioRange$	Avg. node connectivity (n_c)	Position error (% $RadioRange$)			Computational complexity
						Mean	Median	RMS	
MDS-MAP(C)†	44	0.30	0.09	0.43	–	–	–	71.7%	$O(n^3)$
dwMDS †	44	0.30	0.09	0.43	–	–	–	41.3%	$O(nn_cL)$
	44	0.30	0.09	0.61	–	–	–	26.7%	
Dist. SOCP	44	0.30	0.09	0.43	13.0	27.6%	29.2%	30.7%	$O(nn_cL)$
	44	0.30	0.09	0.61	24.6	24.1%	23.2%	26.4%	
MDS-MAP(C)‡	200	0.05	0.05	0.15	12.2	–	17%	–	$O(n^3)$
MDS-MAP(P, R)‡	200	0.05	0.05	0.15	12.2	–	6%	–	$O(nn_c^3) + O(n^3)$
Dist. SOCP	200	0.05	0.08	0.22	21.4	11.6%	10.5%	12.7%	$O(nn_cL)$

$$U_j := \begin{bmatrix} 0.5 & 0^{1\times(2n_i+2)} & 0^{1\times n_i} & 0.5(1_{ij}) \\ -0.5 & 0^{1\times(2n_i+2)} & 0^{1\times n_i} & 0.5(1_{ij}) \\ 0 & 1_{i1} & 0^{1\times n_i} & 0^{1\times n_i} \\ 0 & 1_{i2} & 0^{1\times n_i} & 0^{1\times n_i} \end{bmatrix}, j \in N_A(i)$$

where $\mathbf{1_{ij}}$ is a row vector of length n_i with a 1 corresponding to the variable t_{ij} and 0's elsewhere. Similarly, $\mathbf{1_{i1}}$, $\mathbf{1_{i2}}$ are row vectors of length $(2n_i + 2)$ with a 1 corresponding to the coordinates of $x_i = (x_{i1}, x_{i2})$ and 0's elsewhere.

Define:

$$S := \begin{bmatrix} S_1 \\ \vdots \\ S_{n_i} \end{bmatrix}, r := \begin{bmatrix} r_1 \\ \vdots \\ r_{n_i} \end{bmatrix}, U := \begin{bmatrix} U_1 \\ \vdots \\ U_{n_i} \end{bmatrix}$$

$$t := \begin{bmatrix} t_1 \\ \vdots \\ t_{n_i} \end{bmatrix}, y := \begin{bmatrix} t_i^{'} \\ \mathbf{x} \\ \mathbf{t_{ij}} \\ \mathbf{y_{ij}} \end{bmatrix}$$

where $\mathbf{x} = \begin{bmatrix} x_{i1} & x_{i2} \end{bmatrix}^{\mathbf{T}}_{\mathbf{(2n_i+2)\times 1}}$, $\mathbf{t_{ij}} = \begin{bmatrix} t_{ij} \end{bmatrix}_{\mathbf{n_i \times 1}}$ and $\mathbf{y_{ij}} = \begin{bmatrix} y_{ij} \end{bmatrix}_{\mathbf{n_i \times 1}}$. Then the conic constraints can be expressed as $r - S^T y \in \text{Qcone}^2 \times \ldots \times \text{Qcone}^2$, and $t - U^T y \in \text{Qcone}^4 \times \ldots \times \text{Qcone}^4$, where the Cartesian product is taken over n_i cones. ($\text{Qcone}^k = \{(x, y) \in R \times R^{k-1} : x \geq \|y\|\}$).

Defining:

$$\tilde{b} := -\begin{bmatrix} 0 \\ 0^{2(n_i+1)\times 1} \\ 1^{n_i \times 1} \\ 0^{n_i \times 1} \end{bmatrix}, \tilde{A} := \begin{bmatrix} L_i \\ S \\ U \end{bmatrix}^T, \tilde{c} := -\begin{bmatrix} 1 \\ r \\ t \end{bmatrix},$$

problem (5) can be written in the form (6) with $b = \tilde{b}$, $A = \tilde{A}$, $c = \tilde{c}$ and K being the Cartesian product of all the cones.

REFERENCES

[1] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Commun. ACM*, vol. 47, pp. 34–40, June 2004.

[2] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," in *Computer*, vol. 34. IEEE Computer Society Press, Aug. 2001, pp. 57–66.

[3] J. Caffery and G. L. Stuber, "Overview of radiolocation in CDMA cellular systems," *IEEE Commun. Mag.*, vol. 36, pp. 38–45, Apr. 1998.

[4] S. Srirangarajan, A. H. Tewfik, and Z.-Q. Luo, "Distributed sensor network localization with inaccurate anchor positions and noisy distance information," in *Proc. ICASSP 2007*, Apr. 2007, pp. III521–III524.

[5] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE Pers. Commun.*, vol. 7, pp. 28–34, Oct. 2000.

[6] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE Infocom*, vol. 3, Apr. 2001, pp. 1655–1663.

[7] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.

[8] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol. 2, p. 3964, Feb. 2006.

[9] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc sensor network localization," in *Proc. Int. Symp. on Information Processing in Sensor Networks (IPSN)*. Springer Verlag, Apr. 2004, pp. 46–54.

[10] 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," *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[11] 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*, Jan. 2006, pp. 69–84.

[12] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, "Spaseloc: an adaptive subproblem algorithm for scalable wireless sensor network localization," *SIAM J. Optimization*, vol. 17, pp. 1102–1128, Dec. 2006.

[13] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sensor Networks*, vol. 2, pp. 188–220, May 2006.

[14] P. Tseng, "Second-order cone programming relaxation of sensor network localization," *SIAM J. Optimization*, vol. 18, no. 1, pp. 156–185, Feb. 2007.

[15] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, May 2006, pp. 250–261.

[16] H. Lim and J. C. Hou, "Localization for anisotropic sensor networks," in *Proc. IEEE Infocom*, vol. 1, Mar. 2005, pp. 138–149.

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

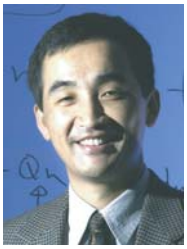
[18] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[19] J. Strum, "Using SeDuMi 1.02, A Matlab Toolbox for Optimization Over Symmetric Cones (Updated for Version 1.05)," Oct. 2001.

Seshan Srirangarajan (S'05-M'08) received the B.E. degree from University of Mumbai, India in 2001, and the M.S. and Ph.D. degrees from University of Minnesota in 2005 and 2008, respectively, all in Electrical Engineering. He is currently a Research Fellow with Intelligent Systems Center at Nanyang Technological University, Singapore. In 2005-06, he was with the Wireless Technologies Group at Honeywell Technology Center, Minneapolis, Minnesota, as an intern. His research interests include ranging and positioning in wireless networks, sensor networks, wireless communication and signal processing.

Ahmed H Tewfik (F'96) received his B.Sc. degree from Cairo University, Egypt, in 1982 and his M.Sc., E.E. and Sc.D. degrees from MIT, Cambridge, MA, in 1984, 1985 and 1987 respectively. Dr. Tewfik worked at Alphatech, Inc., Burlington, MA in 1987. He is the E. F. Johnson professor of Electronic Communications with the Department of Electrical Engineering at the University of Minnesota. He served as a consultant to several companies, and has worked with Texas Instruments and Computing Devices International. From 1997 to 2001, he was the President and CEO of Cognicity, Inc., an entertainment marketing software tools publisher that he co-founded, on partial leave of absence from the University of Minnesota. His current research interests are in genomics and proteomics, audio signal separation, wearable health sensors, brain computing interface and programmable wireless networks.

Prof. Tewfik is a Fellow of the IEEE. He was a Distinguished Lecturer of the IEEE Signal Processing Society in 1997-1999. He received the IEEE third Millennium award in 2000. He was elected to the board of governors of the IEEE Signal Processing Society in 2005. He was invited to be a principal lecturer at the 1995 IEEE EMBS summer school. He was awarded the E. F. Johnson professorship of Electronic Communications in 1993, a Taylor faculty development award from the Taylor foundation in 1992 and an NSF research initiation award in 1990. Prof. Tewfik delivered plenary lectures at several IEEE and non-IEEE meetings and taught tutorials on bioinformatics, ultrawideband communications, watermarking and wavelets at major IEEE conferences. He was selected to be the first Editor-in-Chief of the IEEE SIGNAL PROCESSING LETTERS from 1993 to 1999. He is a past associate editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, was guest editor of special issues of that journal, the IEEE TRANSACTIONS ON MULTIMEDIA and the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He is currently an Associate Editor of the EURASIP JOURNAL ON BIOINFORMATICS AND SYSTEMS BIOLOGY. He also served as the president of the Minnesota chapters of the IEEE signal processing and communications societies from 2002 to 2005.



Zhi-Quan (Tom) Luo (F'07) received his B.Sc. degree in Applied Mathematics in 1984 from Peking University, Beijing, China. Subsequently, he was selected by a joint committee of American Mathematical Society and the Society of Industrial and Applied Mathematics to pursue Ph.D study in the United States. After an one-year intensive training in mathematics and English at the Nankai Institute of Mathematics, Tianjin, China, he entered the Operations Research Center and the Department of Electrical Engineering and Computer Science at MIT, where he received the Ph.D degree in Operations Research in 1989. From 1989 to 2003, Dr. Luo held a faculty position with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada, where he eventually became the department head and held a Canada Research Chair in Information Processing. Since April 2003, he has been with the Department of Electrical and Computer Engineering at the University of Minnesota (Twin Cities) as a full professor and holds an endowed ADC Chair in digital technology. His research interests lie in the union of optimization algorithms, data communication and signal processing.

Prof. Luo serves on the IEEE Signal Processing Society Technical Committees on Signal Processing Theory and Methods (SPTM), and on the Signal Processing for Communications (SPCOM). He is a co-recipient of the 2004 IEEE Signal Processing Society's Best Paper Award, and has held editorial positions for several international journals including JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS, SIAM JOURNAL ON OPTIMIZATION, MATHEMATICS OF COMPUTATION, and IEEE TRANSACTIONS ON SIGNAL PROCESSING. He currently serves on the editorial boards for a number of international journals including MATHEMATICAL PROGRAMMING and MATHEMATICS OF OPERATIONS RESEARCH.