Convex Position Estimation in Wireless Sensor Networks

Lance Doherty[1], Kristofer S. J. Pister, Laurent El Ghaoui
Dept. of Electrical Engineering and Computer Sciences – University of California, Berkeley
497 Cory Hall; Berkeley, CA 94720

***Abstract*— A method for estimating unknown node positions in a sensor network based exclusively on connectivity-induced constraints is described. Known peer-to-peer communication in the network is modeled as a set of geometric constraints on the node positions. The global solution of a feasibility problem for these constraints yields estimates for the unknown positions of the nodes in the network. Providing that the constraints are tight enough, simulation illustrates that this estimate becomes close to the actual node positions. Additionally, a method for placing rectangular bounds around the possible positions for all unknown nodes in the network is given. The area of the bounding rectangles decreases as additional or tighter constraints are included in the problem. Specific models are suggested and simulated for isotropic and directional communication, representative of broadcast-based and optical transmission respectively, though the methods presented are not limited to these simple cases.**

***Index terms*— position estimation, location, sensor networks, convex optimization**

I. Introduction

The maturing of integrated circuitry, microelectromechanical systems (MEMS) and communication theory has fomented the emergence of wireless sensor networks and precipitated the economic and computational feasibility of networks of hundreds or thousands of self-sufficient sensor nodes. Each node has the ability to sense elements of its environment, perform simple computations, and communicate either among its peers or directly to an external observer. Larger node numbers allow for sensing over larger geographical regions with greater accuracy than previously possible.

The work in this paper is principally motivated by the Smart Dust project, aiming to scale sensing communication platforms down to cubic millimeter volume [1]. The most promising methods for short and long range peer-to-peer communications are RF and optical media [2], respectively. Progress is being made to integrate both of these media at the chip level [3], while macroscopic sensor node systems have been demonstrated [4]. Ad hoc routing protocols have been developed – information theoretic bounds are explored in [5] and protocols are detailed in [6], [7].

In a network of thousands of nodes, it is unlikely that the designer will determine the position of each node. In an extreme case, nodes may be dropped from the air and scattered about an unknown area. To process sensor data, however, it is imperative to know where the data is coming from. Nodes could be equipped with a global positioning system (GPS) to provide them with knowledge of their absolute position, but this is currently a costly (in volume, money, and power consumption) solution. Instead, positional information can be inferred from connection-imposed proximity constraints. In this model, only a few nodes have known positions (perhaps equipped with GPS or placed deliberately) and the remainder of the node positions are computed from knowledge about communication links. A less general attempt at solving for node positions relative to beacons is proposed by Estrin *et al*. [8] while hardware-implemented systems are described in [9] and [10].

This paper describes feasible solutions to the position estimation problem using convex optimization. If one node can communicate with another, a proximity constraint exists between them. As a physical example, if a particular RF system can transmit 20m and two nodes are in communication, their separation must be less than 20m. These constraints restrict the feasible set of unknown node positions. Only planar networks are considered, but augmenting the methodology to 3-D is straightforward. In summary as illustrated in Figure 1:

Given: positions of solid nodes

Find: a possible position for each open node

Subject to: proximity constraints imposed by known connections

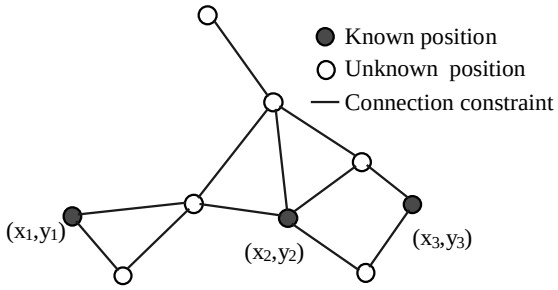


Figure 1. Graph illustrating data and variables as vertices, constraints as edges.

Formally, the network is a graph with n nodes at the vertices (each node having a Cartesian position) and with bi-directional communication constraints as the edges. Positions of the first m nodes are known (x_1, y_1, ... x_m, y_m,) and the remaining n-m positions are unknown. The feasibility problem is then to find (x_{m+1}, y_{m+1}, ... x_n, y_n) such that the proximity constraints are satisfied. Note that there are constraints among open nodes though their positions are unknown. Connections that are not reported are not detrimental to the performance of the algorithm.

The position estimation methodology developed herein requires centralized computation. Namely, all nodes must communicate their connectivity information to a single computer to solve the optimization problem. The feasibility of such a paradigm has been demonstrated by Culler *et al.* [11], generating an RF ad hoc network with the ability to relay

[1] ldoherty@bsac.eecs.berkeley.edu

connection statistics to a centralized PC. This system consists of a custom operating system running on sensor nodes and suggests a concrete example of a sensor network in the context of this paper. We focus exclusively on the position estimation aspect and no further consideration is given to communication protocols though bandwidth constraints may be the fundamental limitation to sensor network size.

II. Mathematical Formulation

Efficient polynomial-time algorithms based on interior-point methods exist for solving linear programs [12] and semidefinite programs [13]. A linear program (LP) is a problem of the form:

$$\begin{aligned} &\textbf{Minimize} \quad \boldsymbol{c^T x} \\ &\textbf{Subject to:} \ \boldsymbol{Ax < b} \end{aligned} \tag{1}$$

Geometrically, this amounts to minimizing a linear function over a polyhedron.

A generalization of the LP is the semidefinite program (SDP) of the form:

$$\begin{aligned} &\textbf{Minimize} \quad \boldsymbol{c^T x} \\ &\textbf{Subject to:} \ \boldsymbol{F(x) = F_0 + x_1F_1 + \ldots + x_nF_n < 0} \\ &\qquad\qquad \boldsymbol{Ax < b} \\ &\qquad\qquad \boldsymbol{F_i = F_i^T} \end{aligned} \tag{2}$$

The first inequality represents a matrix inequality on the cone of positive semidefinite matrices, i.e. the eigenvalues of $F(x)$ are constrained to be nonpositive. This is known as a linear matrix inequality (LMI). Again the objective function must be linear for SDP. Constraints can be stacked in either method. SDP will be sufficient to solve all the numerical problems in this paper though LP will be used where applicable because of superior computational efficiency.

In two dimensions, each node has a position (x,y). For position estimation, a single vector with all the positions is formed:

$$\boldsymbol{x} = [x_1 \ y_1 \ \ldots \ x_m \ y_m \ldots \ x_{m+1} \ y_{m+1} \ \ldots \ x_n \ y_n]^{\mathrm{T}}$$

The first m entries are fixed as data and the remaining n-m are computed by the algorithm.

In general, efficient computational methods are available for most convex programming problems. Geometrically, a convex set is one for which any two points in the set can be connected with a line entirely contained in the set. Convex constraint models for RF and optical communication systems are presented in the following section. Other convex constraints are also computable with the same algorithms.

The solution methods are not approximate; providing that the experimenter believes in the validity of the constraint model, position estimation as developed herein is the best that can be accomplished. Results indicating performance below the desired level for a particular application reflect limitations imposed by uncertainty in the constraint models, not by the position estimation methodology.

III. Convex Constraint Models

A. Connections as convex constraints

Providing that the connectivity of the network can be represented as a set of convex position constraints, the mathematical methods outlined in the preceding section can be utilized to generate feasible positions for all the nodes in the network. It is sufficient to consider connection constraints individually as both programming methods allow for constraints to be collected into a single problem. The question becomes: "given the position of node A, what is the set of possible positions for node B?" The remainder of this section is devoted to two models of this feasible set.

B. Radial constraint – RF communication

The RF transmitter of a wireless sensor node can be modeled as having a rotationally symmetric range as illustrated in Figure 4a. This is not an accurate physical representation of what is often a highly anisotropic and time-varying communication range, but a circle that bounds the maximal range can always be used. Furthermore, the proceeding methods apply equally well to ellipses without increased complexity should it become evident that an elliptical communication model is more relevant.

In this symmetric model, a connection between nodes can be represented by a 2-norm constraint on the node positions. Specifically, for a maximum range R and node positions a and b, the equivalent LMI is given by:

$$\|a-b\|_2 \le R \Rightarrow \begin{bmatrix} I_2 R & a-b \\ (a-b)^T & R \end{bmatrix} \ge 0 \tag{3}$$

For a two-dimensional problem, using Schur complements [14] transforms the quadratic inequality into an LMI with a 3x3 matrix in (3) with I_2 representing the two-dimensional identity. Multiple LMIs can be stacked in diagonal blocks to form one large SDP for the entire network. Hence, each bi-directional proximity constraint contributes one convex 3x3 LMI to the system.

A variant of this problem is to use the exact distance between nodes as the outer distance bound in the above LMI:

$$\begin{bmatrix} I_2 r_{ab} & a-b \\ (a-b)^T & r_{ab} \end{bmatrix} \ge 0; \quad r_{ab} \text{ given} \tag{4}$$

For emphasis, in the former case all constraints are bounded with a circle with the same radius R, in the latter case, each constraint is assigned the smallest radius r_{ab} possible (Figure 2). Physically, transmitters varying their output power during an initialization phase could obtain an estimate of r_{ab}. If a connection is first obtained at a power P_o, the receiver calculates the maximum possible separation for reception at P_o. This maximum separation $r_{ab} < R$ can be used to determine

a tighter upper bound on each individual connection in the network.

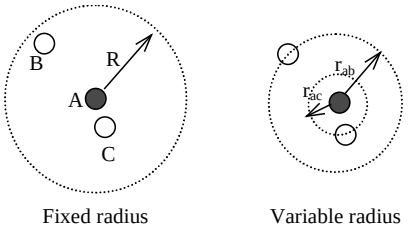


Figure 2. In the fixed radius case, both constraints AB and AC are circles of radius *R*. In the variable radius case, constraint AB has a radius r_{ab} and constraint AC has a radius r_{ac}. There would also be a constraint BC in each scenario.

Note that neither of the following are convex constraints:

$$\|a-b\|_2 = r_{ab}; \quad \|a-b\|_2 > R \tag{5}$$

The latter would be useful in "pushing away" nodes that are not connected in the algorithm as in Figure 3. This constraint is not physically realistic either – nodes within a certain range may not be able to communicate due to a physical barrier or transmission anisotropy. Even more precision could be obtained if some lower bound on separation was known and the former constraint in (5) was formulated as a set of robust convex constraints.

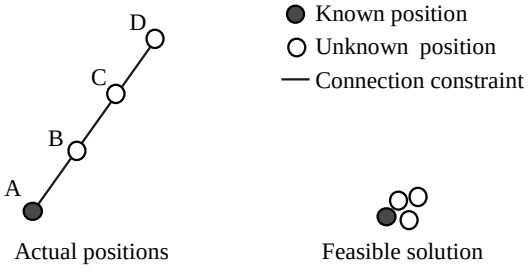


Figure 3. There is no mechanism in the radial constraint model for bounding nodes away from known positions. As illustrated, the entire node chain could feasibly collapse to a point as the unknown nodes B, C and D lie in a feasible set of circles of radii *R*, 2*R*, and 3*R* centered at A.

C. Angular constraint – optical communication

Consider sensor nodes with laser transmitters and receivers that scan through some angle. The receiver rotates its detector coarsely until a signal is obtained, and then finely to get the maximum signal strength. By observing the angle at which the best reception occurs, we can form an estimate of the relative angle to the transmitter and a vague estimate of the maximum distance between them. This results in a cone (triangle in 2D) for the feasible set as in Figure 4b. This cone can be expressed as the intersection of three half-spaces – two to bound the angle and one to place a distance limit. The intersection of half-spaces is still an LP.

A node connected to plural neighbors will have a cone for each neighbor pointed in the appropriate direction, but all cones have the same half-angle and length.

D. Other convex constraints

Any combination of the SDP and LP constraints can be used to define individual feasible position sets. Some study was devoted to a quadrant detector scheme (Figure 4c) involving one LMI and two scalar linear constraints. In a more general case, for a trapezoid with variable angles and width (Figure 4d), four linear constraints can approximate a segment of an annulus that might represent uncertain knowledge of both position and angle. There are other non-LMI constraints that may also prove useful.

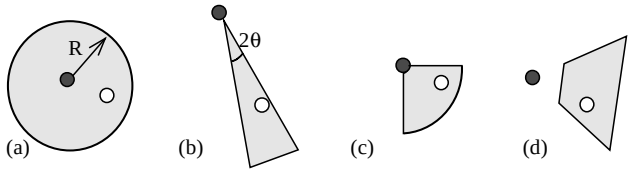


Figure 4. Geometrical interpretation of single constraints. Given that the solid and open nodes are connected, the open node must lie in the shaded region anchored by the solid node position. Constraint shown in (a) radial, (b) angular, (c) quadrant and (d) trapezoid. The outer bound in (b) is optional.

E. Combining Individual Constraints

Node positions in the network are often constrained by connections to several other nodes. Satisfying plural constraints means that the feasible set becomes the intersection of the individual constraint sets, necessarily making the feasible region smaller with each added constraint as in Figure 5. This decreasing area is precisely the mechanism behind the position estimation. The intersection of convex sets is itself a convex set, so our search methodology continues to be sufficient.

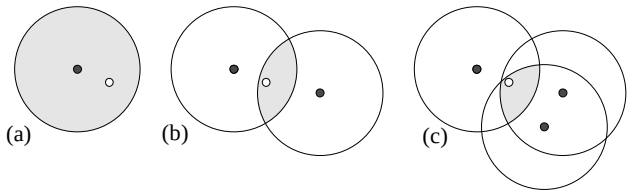


Figure 5. Combination of radial constraints. The shaded region represents the feasible set for the light node, constrained by the dark node positions. From (a)-(c), the intersected constraints yield progressively smaller feasible sets.

Though only connections between known positions and unknown positions are illustrated in Figure 5, unknown-unknown connections are equally important to the problem solution. In this case, both *a* and *b* are variables in the SDP or LP. All unknown positions are solved simultaneously using a single global program.

TABLE 1
Summary of constraint types.

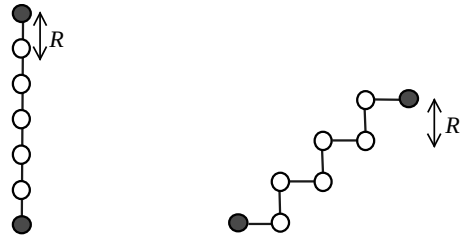
Method	Allowable angle	Allowable distance	Linear inequalities
Fixed radial	$0 - 2\pi$	Fixed	1 LMI
Variable radial	$0 - 2\pi$	Variable	1 LMI
Angular	Variable	Fixed	3 scalar
Quadrant	One quadrant	Fixed	1 LMI, 2 scalar
Trapezoid	Variable	Variable	4 scalar

IV. SIMULATION

A. Computation Time

All simulation is computed on an AMD K6-2 400 MHz processor with 64 MB of RAM. Code is executed from the MATLAB environment using the Mosek Optimization Toolbox [15] to solve the LP and SDP problems. When restricted to a circular region, the radial case can be solved as a second-order cone problem (SOCP). This software uses interior-point methods for efficient solution of both LPs and SOCPs. Sparse arrays are used to store variables and connectivity information throughout.

As an experiment to determine computation time and scaling, two toy problems are solved: one for the radial constraint method and one for angular constraints (Figure 6). The first consists of n nodes arranged in a vertical chain separated by a distance R with only the top and bottom nodes in known positions. In the radial case, the only feasible solution is the actual solution. The second places the n nodes in a staircase pattern, again with the first and last nodes in known positions. In the angular case, the feasible solution is very close (and dependent on theta) to the actual solution. Every unknown node is connected to exactly two other nodes.



(a) Radial toy problem (b) Angular toy problem

Figure 6. Node positions for toy problems. Shaded nodes at the end of the chains are known, all other nodes have unknown positions.

In all cases, the number of fixed nodes, $m = 2$, while the middle node positions are estimated. Computation times are shown in Figure 7 and include only the time spent in the solver, not the time to simulate the network connectivity.

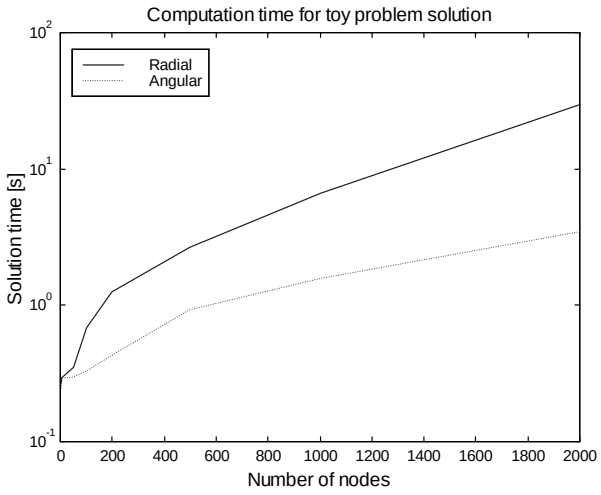


Figure 7. Computation time in seconds required to solve the SOCP and LP toy problems.

Over the range of the presented data, the SOCP scales better than $O(k^3)$ and the LP scales better than $O(k^2)$ where k is the number of connections.

B. Network simulation

Unless otherwise specified, networks used in the simulations were formed by placing nodes randomly and uniformly in a square region of side length $10R$. 200 nodes are placed, then connectivity is determined by examining each pairwise distance; if the distance between two nodes is less than R, the nodes are labeled as connected. Finally, the largest connected subnetwork of the 200 node network is extracted and the node labels are randomly permuted as in Figure 8. 10 such networks are used for simulation, the average node number is 194 and the average connectivity of each node is 5.7.

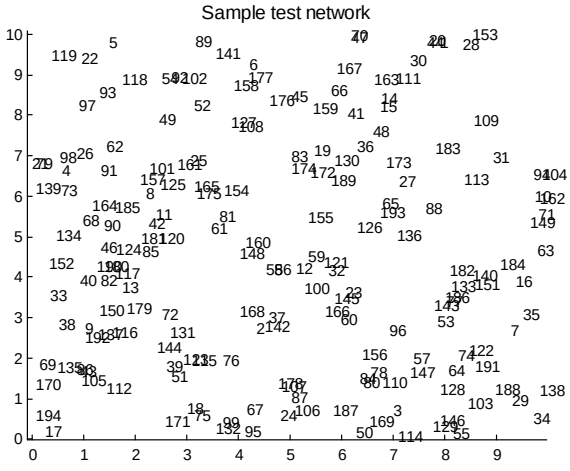


Figure 8. Test network generated from 200 randomly placed nodes. Numbers indicate node labels and position.

C. Performance metric

As mentioned in section II, both the SDP and LP admit linear objectives ($c^T x$) exclusively. However, there is no readily apparent linear objective that would provide any sort of "optimal" solution to the placement problem. Instead, the objective function is left blank in the solver. This has the effect of selecting a *random* feasible point $\boldsymbol{x}_{est} = (x_{est}, y_{est})$ from the solution space – this point represents a (n-m) entry set of (x,y) pairs, one for each unknown position. The most precise statement of a node's position that can be made is that the node lies somewhere in the intersection of the allowable regions. Providing that these regions are small enough, finding a feasible position for all the nodes may be close enough for the required estimation.

The performance of the algorithm is defined as the mean error from the computed to the actual unknown positions. This mean error provides a measure of the size of the feasible set.

$$error = \frac{1}{n-m}\sum_{i=m+1}^{n}\left\| x_{est}^{i} - x_{real}^{i} \right\|_2 \qquad (6)$$

The same metric is utilized for both the radial and angular problems. While it is not the most intuitively satisfying for the angular case, the problem we are interested in is position estimation and as such the deviation of actual position is still the best variable for minimization.

D. Bounding the feasible set

To effectively exploit the utility of the objective function $c^T x$, the algorithms can be run multiple times. This provides a mechanism for bounding the feasible set with a rectangle parallel to the axes. The computation proceeds as follows for *each* unknown position k:

1) Set $c_{2k-1} = 1$ and all other $c_i = 0$ for c in (1) or (2)
2) Solve the original SDP/LP → yields x^k_{min}
3) Set $c_{2k-1} = -1$ and all other $c_i = 0$
4) Solve original SDP/LP → yields x^k_{max}
5) Repeat steps 1-4 for c_{2k} to get y^k_{min} and y^k_{max}

This procedure defines the smallest such rectangle that bounds the feasible set as in Figure 9. By selecting the center of this rectangle as the most likely solution, we can expect some improvement in mean error over the randomly selected case.

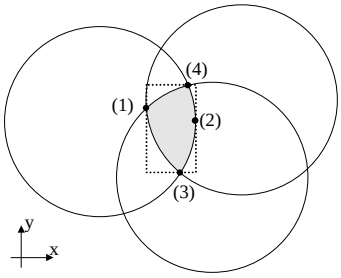


Figure 9. The shaded region represents the feasible set for this problem. The procedure finds points 1-4 that define the tight rectangular upper bound shown in dots. This rectangle runs parallel to the axes shown.

For the price of a 4(n-m) times increase in the number of problems solved, an increase in estimation performance and an outer bound on the solution is obtained.

Another possibility is to find the minimum measure elliptical bound on the solution space for each unknown position as discussed in [14]. This does not, in general, provide a tight upper bound due to the problem relaxation, but may provide numerically similar results requiring the solution of a single SDP for each unknown position instead of four. However, the problem no longer falls into the simpler SOCP framework.

V. Radial Constraint Results

A. Comparison of two radial constraint methods

The performance difference between the fixed radius and variable radius RF location methods is analyzed and explained in this section. Using an n node network, the following test is performed:

1) Select node 1 as a known position ($m = 1$)
2) Solve for the remaining n-m unknown positions
3) Compute the mean error for these n-m positions from the actual network with (6)
4) Increase the number of known positions by 1 (hence decreasing the unknowns by 1, $m = m + 1$)
5) Repeat steps 2-4 until $m = 100$

The trials cease at $m = 100$ as the implementation of position calibration in more than half of the nodes would obviate the intended cost savings. Results of these trials are summarized in Figure 10.

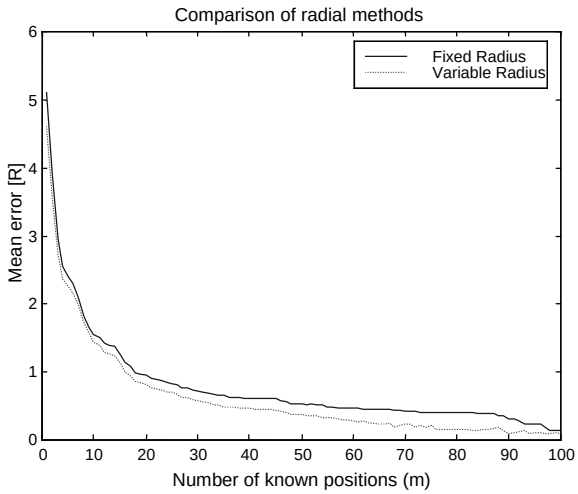


Figure 10. Fixed and variable radius estimation methods with errors averaged over the 10 test networks.

At low values of m, the estimation performs as poorly as a random guess at the node locations. It is useful to compare the performance against a naïve beacon system in which the environment is covered by a grid of known node positions. If a node is within communication range of the beacon, a random guess within this radius of R would result in a mean error of

2/3 R for the network. For our 10Rx10R network, this performance would require over 100 beacon nodes; this accuracy is achieved with 26 nodes in the variable radius case and 33 for fixed radii with *randomly* chosen known positions.

There is a significant performance increase with the variable radius method, suggesting that efforts to improve distance sensing either by measuring power directly or by modulating the transmission power through a few discrete steps will ameliorate position estimation. The error decreases monotonically with m, the more known positions, the better the performance.

B. Intelligent selection of known nodes

As expounded in section III.B, the radial constraint has no propensity for pushing nodes away. Unknown positions will not be placed beyond the convex hull of the known positions in this estimation methodology. Hence, the best results should be obtained when the known node locations are found on the perimeter of the network, especially at the corners.

In this experiment, the 4 nodes closest to the corners of the test network are chosen as the known positions. Referring to Figure 8, this is the set of nodes {17,34,119,153}. Averaged over the 10 test networks, corner node selection reduces mean error in the variable radius case from 2.4R to 1.2R. Selecting the additional nodes closest to the middle of the external edges for a total of 8 known positions reduces mean error from 1.7R to 0.72R. With 8 known positions placed at the network perimeter, the 100+ beacon network performance is matched.

C. Rectangular bounds

The solution of the radial SDP with a linear objective yields an extremum of one unknown variable. Repeating this procedure 4(n-m) times yields a rectangular bound on all unknown node positions. An example of the outcome for a simple network with 3 known positions is shown in Figure 11.

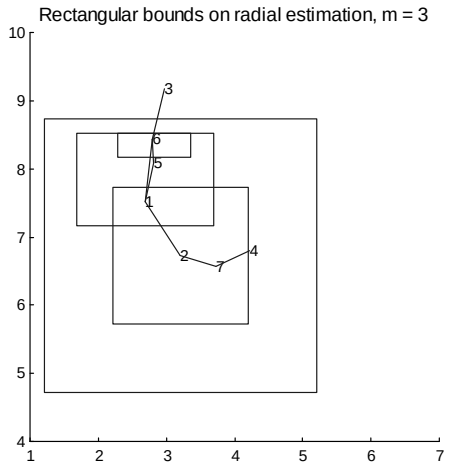


Figure 11. Numbers indicate actual node positions. Rectangles illustrate the uncertainty in the positions of nodes 4 through 7 with node 4 having the largest uncertainty. Lines between nodes indicate connectivity.

The smallest rectangle is associated with node 6, connected to two known positions. Node 5 is connected only to one known position, node 1, but additional constraints imposed by node 6 aid in reducing the feasible position set. The allowable set for node 7 is the circle of radius R = 1 around node 2, as illustrated by the bounding rectangle of side length 2. Finally, node 4 can lie anywhere within the circle of radius 2R = 2 centered at node 2 and has the biggest rectangle.

The outer bounds provide a convenient means of assessing the effect of connectivity on the measure of the feasible position sets. Intuitively, it is expected that nodes with more neighbors will have smaller feasible sets because of the intersection property. To validate this claim, the bounding algorithm is tested with the variable radius case on a single test network with eight perimeter known positions. Results appear in Figure 12.

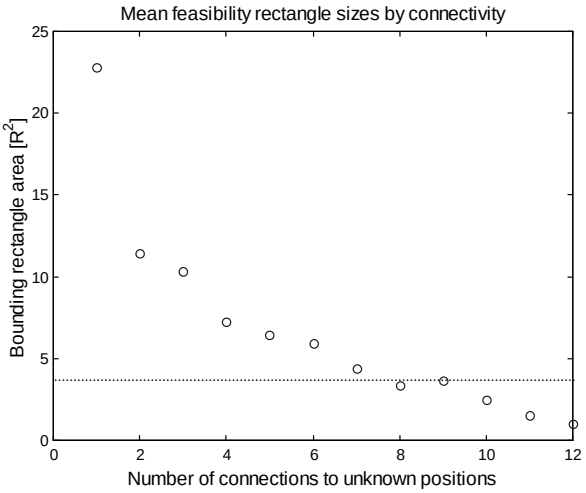


Figure 12. Correlation between connectivity and bounding rectangle area. Only nodes not connected to known positions are considered. The dotted line is drawn at $4R^2$.

As a comparison, recall that nodes with connections to known positions have a rectangle area bounded above by $4R^2$. About eight connections to unknown positions are required to achieve this level of certainty.

The same set of results are used to determine the effect that the distance from a known node has on the precision of the unknown solutions. The raw data for each node is given in Figure 13.

Two effects are evident. Nodes within a distance R from a known position must have a rectangle area less than $4R^2$. Beyond this, there is a full range of rectangle sizes up to a distance of 3R. At this point, the only nodes with such distance properties are the ones situated at the center of the network, those with the highest connectivity. This high connectivity is responsible for the reduction in feasible set area near the right of the plot.

Additionally, the selection of the rectangle centroids for the unknown positions does improve the accuracy of the estimation as speculated. The mean error drops from 0.72R (as given in section V.B) to 0.64R.

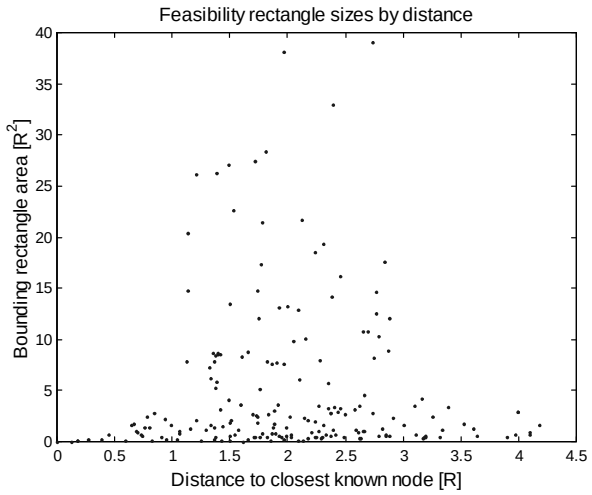


Figure 13. Rectangle sizes for unknown nodes with different minimum distances to known nodes.

VI. Angular Constraint Results

A. Variation in cone half-angle

The same 10 networks and error metric are employed to test the angular estimation model. More information is required from the nodes: not only does the centralized computer need a binary value representing each pairwise connection, but also the angle between each connected pair. Two parameters are varied in the experiments: the half-angle of uncertainty (θ in Figure 4b), and the distance to the outer bound of the cone.

In the first experiment, θ is reduced from $\pi/4$ to $\pi/10$ and to $\pi/100$. The first case is an outer approximation to the more complicated quadrant constraint discussed earlier. Again, the number of known positions is increased from 1 to 100 and the mean error is computed over the 10 test networks. As anticipated, the smaller individual constraints lead to better position estimates as plotted in Figure 14.

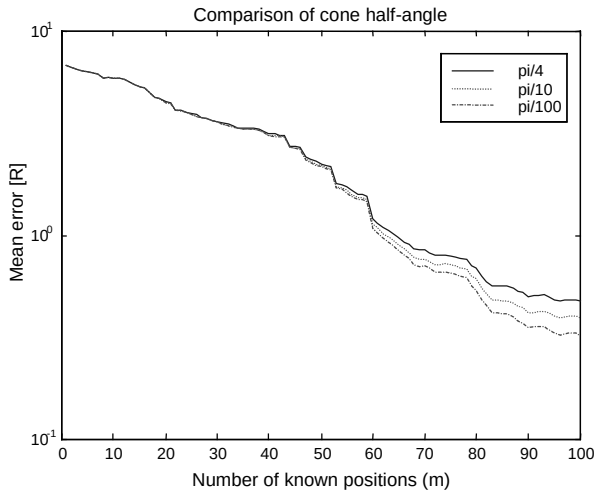


Figure 14. The half-angle of the connectivity cones is varied and error is averaged over the 10 test networks.

The three curves follow a similar pattern as they use the same node ordering during solution of the test networks. As *m* increases, the mean error will decrease in jumps when an "important" node becomes known. Such nodes are often located at the perimeter of the network or serve to link two parts of the network. The overall error does not increase as quickly as the area of the constraint cone.

B. Variation in cone outer bound

To determine the sensitivity of the results to the uncertainty in the length of the cone, the outer bound is varied in this experiment as shown in Figure 15.

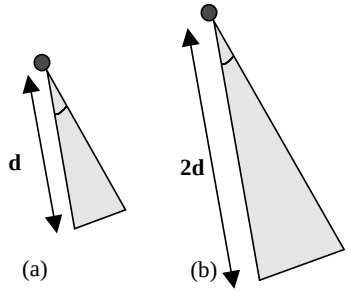


Figure 15. The outer bound in (b) is twice as long as that in (a) while the half-angle remains constant.

The connectivity of the network is determined using the same distance as previously; the nodes have no more connections than before, but the positional uncertainty of neighboring nodes has increased. If no distance information was available, the cone would have an infinite length. Results are shown in Figure 16 for increased outer bound lengths.

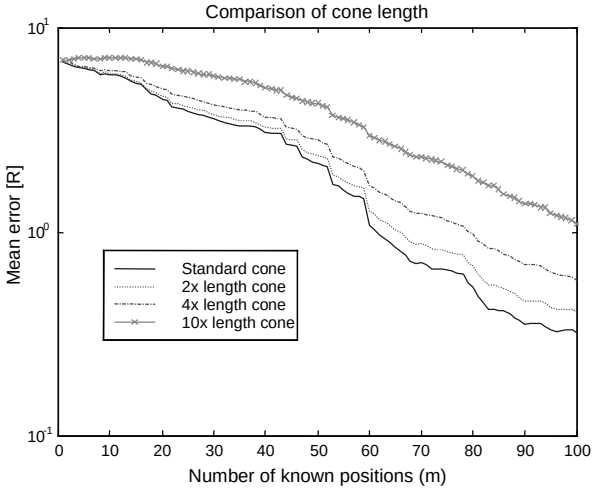


Figure 16. Cone length is increase beyond the connection radius *R*. The mean error is an average over the 10 test networks. A half-angle of $\pi/100$ is used in all three trials.

The cone length of 10*R* essentially robs the solver of any distance information as the simulation environment has a side length of 10*R*. Even with many known positions, this large cone constraint does not perform well, suggesting that some distance information is a necessity in a practical application. At low values of *m*, the cone length does not matter as the position estimation is little better than a random guess. The

parameters cause divergence in the results as the known nodes increase and smaller intersections of feasible sets are obtained. Again, the overall error increases more slowly than the area of the cone.

C. *Increased node density*

By artificially increasing the connectivity distance, increased node density can be simulated. Using the same 10 networks, the connectivity matricies are reconstructed using thresholds of 1.25*R*, 1.5*R*, 1.75*R*, and 2*R*. From the original mean of 5.8 connections/node, the connectivity rises to 8.5, 12.2, 16.1 and 20.4. This grows approximately as the circular area of connectivity; the slight discrepancy is likely due to network edge effects. This increased connectivity betters the position estimation as seen in Figure 17.

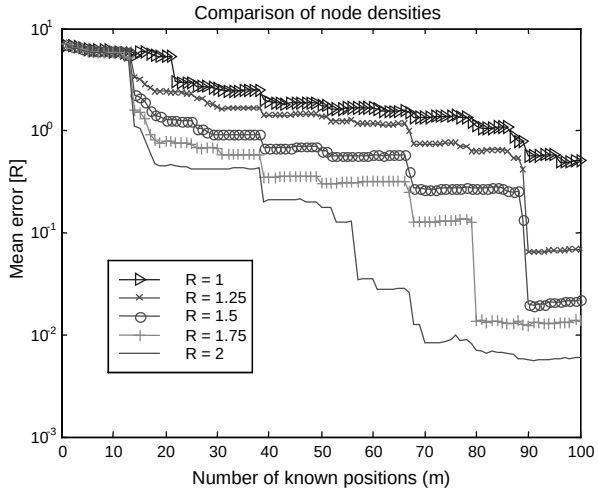


Figure 17. Different node density based on a single test network. Error levels decrease as node density increases.

The jumps in the mean error are more evident in this trial as only one test network is used. The labels of the "important" nodes can be read off from the plot. Higher connectivity results in more such breakthrough nodes. At 50% known nodes, the four-fold increase in connectivity results in a two order of magnitude decrease in position estimation error.

VII. Conclusions

A methodology for formulating a sensor network position estimation problem as a linear or semidefinite program is proposed. This formulation is based on connectivity and pairwise angles between nodes. It was shown that a rapid solution of the problem for networks with several hundred nodes is possible, and that this technique is directly extensible to networks of thousands of nodes.

In the case of radially constrained connections, using a variable radius constraint improves performance over a globally fixed radius. The placement of known positions around the perimeter of the network results in small errors with as few as four known positions. While requiring significant computational overhead, rectangles placed around the feasible set not only bound the possible positions of the unknown nodes, but also provide a better estimate.

For angle constrained conections, decreasing uncertainty through a reduction in half-angle or cone length reduces the mean position estimation error over the network. In a practical optical communication scenario, the angle might be known precisely while a conservative distance approximation (long outer cone bound) would be required. Simulation shows that neglecting distance information entirely adversely impacts performance. Increases in graph connectivity improve performance dramatically, but would require significant increases in communication in the network to transmit all the required connectivity information to the central computer. Obtaining the connectivity information will require a number of messages linear in the average network connectivity and the solution of the problem will scale polynomially as appropriate for the LP or SOCP formulation.

Results for the angular and radial methods should not be compared directly as different numerical solvers with disparate initializations and random objective functions are used.

VIII. Applications and Future Work

A. *Tracking through the sensor network*

A specific application of the procedures described is tracking of an object through the sensor network. The sensing radius can be modeled as in the radial constraint case. If multiple nodes can sense the object, the same set intersection methods via SDP can be utilized to estimate the object's position and provide an upper bound. This is a problem with only one unknown – the position of the tracked object – and n known node positions. The solution should hence be rapid and possibly simple enough to accomplish using the microprocessor of a sensor node. Of course, this can be extended to track k objects concurrently, analogous to the k unknown node positions developed previously.

Acoustical data is used by Yao *et al.* [16] to locate an object using sensors. Yao's method uses more specific (phase-related) data and provides more precise results.

B. *Hierarchical solution for large networks*

As networks scale beyond the demonstrated 2000 nodes, the problems (particularly the radial constraint method) become computationally intensive. This is an inauspicious result for scaling to networks of hundreds of thousands of nodes. We propose two possibilities: limit the number of constraints on each node or solve the problem hierarchically.

The first option is to impose an upper bound on the number of constraints that will be considered for each node. A performance level can be chosen such that the rectangular outer bounds average to a certain value by designing the network for a particular node density. In essence, we keep the same number of connections in our problem while dividing them up among more nodes. For example, a halving of the mean connectivity of a network allows for twice as many nodes to be positioned for the same number of constraints. This does not, however, provide for scaling the solution

through orders of magnitude.

The second option is to first divide a large network into smaller subnetworks based on connectivity data – nodes connected to one another will likely be members of the same subnetwork. Position estimation can be carried out for each member of the subnetwork based on an unknown centroid of this region. Following the individual estimations, the subnetwork centroids can be abstracted to nodes in the larger network and placed accordingly with another iteration of position estimation. This method mirrors Lagrangian Decomposition for the solution of large optimization problems on multiple processors. With plural hierarchical steps, this computational method scales linearly to arbitrarily large problems.

C. *Implementing continuous distributions*

The feasible sets are currently defined as binary fields – either a node is permitted to be at a position *(x,y)* or it is not. An extension of this theory could be made to continuous distributions. Based on a certain power reading, there may be a Gaussian distribution of the most likely distance between nodes. Solution of the placement problem would then give probabilistic distributions for all unknown node positions. From this, the most likely position can be determined as well as confidence intervals.

D. *Combination of angular and radial constraints*

A practical model of a heterogenous system might incoporate both radial and angular constraints in the same network. Some nodes might receive optical signals from precise angles with poor distance information, while others might receive precise distance information in the absence of any directional knowledge. The combination of the two types of constraints could result in a hybrid with better overall performance. Another scenario involves nodes with both RF capabilities and beam steering mechanisms that could combine constraint types at the node level.

E. *Erroneous data management*

There is currently no faculty for detecting erroneous connections. If a proximity constraint is fallaciously reported, the algorithm will, in general, fail. Testing for such an error is as difficult as solving the position estimation problem outright.

F. *Modeling uncertainty in "known" positions*

Throughout this paper, it has been assumed that the "known" node locations have been precisely determined. It is simple to introduce some uncertainty in this model by adding a single convex constraint. For example, suppose that node A is positioned at the origin, uncertain to within a unit distance. By adding a virtual node position at the origin, node V, and connecting this node to node A with a radial constraint of $r_{AV} = 1$, the uncertainty will be accounted for by the global problem solution.

This methodology simultaneously allows for a sensitivity study on the known positions. By varying the uncertainty on the known node positions and measuring the corresponding variation in network error, the importance of precise positioning can be inferred.

ACKNOWLEDGEMENTS

This work was supported by the DARPA/MTO and DARPA/ITO MEMS programs and an NSERC Canada postgraduate fellowship.

REFERENCES

[1] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Mobile Networking for Smart Dust", ACM/IEEE Intl. Conf. on Mobile Computing and Networking, Seattle, WA, Aug. 1999.

[2] M. Last, K. S. J. Pister, "2-DOF Actuated Micromirror Designed for Large DC Deflection", MOEMS '99, Mainz, Germany, Aug 1999.

[3] B. Atwood, B. Warneke, K. S. J. Pister, "Preliminary circuits for Smart Dust", IEEE Southwest Symposium on Mixed-Signal Design, pp. 87-92, San Diego, CA, Feb. 2000.

[4] G. Asada, M. Dong, T. S. Lin, F. Newberg, G. Pottie, W. J. Kaiser, H. O. Marcy, "Wireless integrated network sensors: low power systems on a chip", ESSCIRC '98. Proceedings of the 24th European Solid-State Circuits Conference, The Hague, Netherlands, Sep. 1998.

[5] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", IEEE Transactions on Information Theory, vol. 46, no. 2, pp. 388-404, Mar. 2000.

[6] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols", ACM/IEEE Int. Conf. on Mobile Computing and Networking, Dallas, TX, Oct. 1998.

[7] J. Kulik, W. R. Heinzelman, H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks", ACM/IEEE Int. Conf. on Mobile Computing and Networking, Seattle, WA, Aug. 1999.

[8] N. Bulusu, J. Heidemann, D. Estrin, "GPS-less low cost outdoor localization for very small devices", Technical report 00-729, Computer Science department, University of Southern California, Apr. 2000

[9] J. Hightower, G. Borriello, R. Want, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength", University of Washington CSE Technical Report #2000-02-02, Feb. 2000.

[10] N. B. Priyantha, A. Chakraborty, H. Balakrishnan, "The Cricket Location-Support System", ACM/IEEE Intl. Conf. on Mobile Computing and Networking, Boston, MA, Aug. 2000.

[11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. S. J. Pister, "System architecture directions for networked sensors", Operating Systems Review, vol.34, pp. 93-104, Dec. 2000.

[12] N. Karmarkar, "A new polynomial-time algorithm for linear programming", Combinatorica, vol. 4, pp. 373-395, 1984.

[13] Y. Nesterov, A. Nemirovskii, Interior-Point Polynomial Algorithms in Convex Programming, SIAM, 1994.

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

[15] Mosek Optimization Software, [online] http://www.mosek.com

[16] C. W. Reed, R. Hudson, K. Yao, "Direct joint source localization and propagation speed estimation", 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Phoenix, AZ, Mar. 1999.