Localization (Position Estimation) Problem in WSN

[1] "Convex Position Estimation in Wireless Sensor Networks" by L. Doherty, K.S.J. Pister, and L.E. Ghaoui

[2] "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization" by P. Biswas and Y. Ye

Problem setup and other generalities

In our models, we consider two popular methods for short and long range peer-to-peer communications: RF and optical media. Only planar networks will be considered, but extending the developed localization techniques to 3D is straightforward.

In a network of thousands of nodes, usually densely deployed, it is unlikely that the designer will determine the position of each node. To process sensor data, however, it is necessary to know where the data come from. Equipping every node with a global positioning system (GPS) is currently a costly (in volume, money, and power consumption) solution. Instead, we can estimate node positions relying only on measurements of distances between nodes and other similar connection-imposed proximity constraints. In this model, only a few nodes (anchors) have known positions (perhaps equipped with GPS or placed deliberately) and positions of the remaining nodes are computed from knowledge about communication links. The distance information can be based on time-difference of arrival, received signal strength, and other criteria.

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. A realistic assumption is that there is some degree of error in the distance information.

In a given network of n nodes, we assume that positions of the first m nodes are known ($x_1, y_1, \ldots x_m, y_m$) and the remaining ($n$-$m$) positions are unknown. The feasibility problem is then to find ($x_{m+1}, y_{m+1}, \ldots x_n, y_n$) such that the proximity constraints are satisfied. Note that it is crucial to take into consideration constraints among unknown nodes. Connections that are not reported are not detrimental to the performance of the presented algorithms (of course, our result may be less accurate if some of the connections are not reported).

The position estimation methodology developed in [1] and [2] requires centralized computation. Namely, all nodes must communicate their connectivity information to a single computer to solve the optimization problem.

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.

In [1], we search for feasible solutions to the position estimation problem using convex optimization, Linear Programming (LP) and Semidefinite Programming (SDP), though the optimization part of those is not used to its full extent.
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.

Providing that the constraints are tight enough, simulation illustrates that estimated positions are 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.

In [2], we set up the optimization problem to minimize the error in node positions to fit distance measures, and convex programming techniques are used to solve it. We convert non-convex quadratic distance constraints (not used in [1]) into linear constraints. That results in estimation errors being minimal even when the anchor nodes are not suitably placed within the network or the distance measurements are noisy. Also observable gauges are developed to measure the quality of the distance data or to detect erroneous sensors.

Machinery: LP and SDP

LP solves problems of the form:

Minimize $\boldsymbol{c}^T\boldsymbol{x}$
Subject to: $\boldsymbol{Ax} \leq \boldsymbol{b}$

Geometrically, we're minimizing a linear function over a polyhedron.

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

Minimize $\boldsymbol{c}^T\boldsymbol{x}$
Subject to: $\boldsymbol{F(x) = F_0 + x_1F_1 + \ldots + x_nF_n \leq 0,\ F_i = F_i^T}$
$\boldsymbol{Ax \leq b}$

Efficient polynomial-time algorithms based on interior point methods exist for solving linear programs and semidefinite programs. In general, efficient computational methods are available for most convex programming problems. (It is easy to see that feasible solutions of LP and SDP form convex sets.)

Constraints can be stacked in the both methods. SDP is sufficient to solve all numerical problems that we encounter below, though LP is used whenever possible because of its superior computational efficiency.

For position estimation, we form a single vector with all the positions:
$\boldsymbol{x} = [x_1\ y_1\ \ldots\ x_m\ y_m \ldots\ x_{m+1}\ y_{m+1}\ \ldots\ x_n\ y_n]^T$
The first m entries are fixed as data and the remaining $(n\text{-}m)$ are computed by the algorithm.

The solution methods are not approximate: providing that we believe in the validity of the constraint model, position estimation obtained 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.

It is sufficient to consider connection constraints individually as both programming methods allow for constraints to be collected into a single problem.

Modeling feasible sets: turning connection constraints into those admissible in LP and SDP

Radial constraints – RF communication

The RF transmitter of a wireless sensor node can be modeled as having a rotationally symmetric range. While this is not an accurate physical representation of what is often a highly anisotropic and time-varying communication range, a circle that bounds the maximal range can always be used. Furthermore, the developed methods apply, without increased complexity, to ellipses. So the methods can still be used if it becomes evident that an elliptical communication model is more relevant.

In the rotationally 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, we have $\| a - b \|_2 \leq R$. It is easy to show that this condition is equivalent to:

$$\begin{bmatrix} I_2 R & a-b \\ (a-b)^T & R \end{bmatrix} \geq 0$$

and this can be presented in the SDP constraint form given above.

We can stack the radial constraints in diagonal blocks to form one large SDP for the entire network. (Thus, each proximity constraint contributes one convex 3x3 linear matrix inequality to the system.)

If we know the exact distance r_{ab} between a and b (or, a tighter (a, b)-specific upper bound), we will use it instead of the global upper bound R. Physically, an estimate of r_{ab} can be obtained during an initialization phase by transmitters varying their output power. If a connection is first obtained at a power P_0, the receiver calculates the maximum possible separation for reception at P_0. This maximum separation $r_{ab} < R$ can be used to determine a tighter upper bound on each individual connection in the network.

We note that the following constraints are not convex (and ignored in [1] altogether):

$\| a - b \|_2 = r_{ab}$, $\| a - b \|_2 > R$.

The former one would be very helpful if formulated as a set of robust convex constraints. It is easy to argue that constraints of the latter type are not physically realistic: nodes within a certain range may not be able to communicate due to a physical barrier or transmission anisotropy. (However, those are used in [2] in their generic constraint model.)

What do we miss ignoring the above non-convex constraints? We do not have a mechanism in the radial constraint model for bounding nodes away from known positions. This means the entire network could feasibly collapse to a point. More generally, unknown positions will always be found in the convex hull of the known positions. Hence, we have to be deeply concerned about placing our anchors, with the best results obtained when they are “uniformly distributed” on the convex hull boundary of the network. Such a limitation may be very uncomfortable in certain cases.

Angular constraints – optical communication

Here we 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 fine-tunes to get the maximum signal strength. By observing the best reception angle, we get an estimate of the relative angle and a rough estimate of the maximum distance to the transmitter. This results in a cone (triangle in 2D) for the feasible set. Such a 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 can be expressed as an LP constraint.

We note that any combination of the SDP and LP constraints can be used to define individual feasible position sets. A practical model of a heterogeneous system might incorporate 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 without any directional knowledge. Combining the two types of constraints could result in a hybrid with better overall performance. We may also consider the scenario when nodes are equipped with both RF and optical mechanisms. That would let us combine constraint types at the node level.

Modeling uncertainty in anchor positions

Although we seemingly assume in our models that the anchor node locations are known precisely, it is simple to introduce some uncertainty by adding new convex constraints. For example, suppose that node A is positioned at the origin, uncertain to within a unit distance. By adding a virtual node positioned at the origin, node V, and adding a radial constraint $r_{AV} = 1$, the uncertainty will be accounted for by the global problem solution.
This also allows for a sensitivity study on the anchor positions. By varying the uncertainty on the known node positions and measuring the corresponding variation in the network error (a measure of discrepancy between actual and estimated node positions), we can infer the importance of precise anchor positioning.

LP/SDP objective function issue for our models. Bounding feasible sets.

While we can express nodes proximity constraints in the form admissible by LP and SDP, there is no natural linear objective ($c^T x$) that would provide any sort of "optimal" solution to the localization problem. One option is to leave the objective function blank in the solver. This has the effect of selecting *some* feasible point $\boldsymbol{x}_{est} = (x_{est}, y_{est})$ from the solution space – this point represents a set of (n-m) pairs (x, y), 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 feasible region. Providing that these regions are small enough, computed feasible positions for all the nodes should be close enough to the actual positions.

We define performance of the algorithm as the mean error in the computed node positions:

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

Taking the objective function into use and running the algorithm multiple times lets us bound the feasible sets with rectangles parallel to the axes. Setting vector c as (0, …, 0, ±1, 0, …, 0) we will obtain minimum and maximum feasible values of x and y coordinates of unknown nodes. Selecting centers of the bounding rectangles as the most likely solution, we can expect an improvement in the mean error. (We can, at least, show that the center of a bounding rectangle always belongs to the corresponding feasible set.) Thus, for the (quite high) price of a $4(n\text{-}m)$-fold increase in the number of problems solved, an improvement in estimation performance and an outer bound on the solution are obtained.

Note that it is also possible to find the minimum measure elliptical bound on the solution space for each unknown position. Although that does not, in general, provide a tight upper bound due to the problem relaxation, we may get numerically similar results by solving a single SDP for each unknown position instead of four. However, the problem then would no longer fall into the simple second-order cone problem (SOCP) framework that we can use when dealing with SDP with radial constraints.

Simulation results

Computation time

Applying LP and SOCP solvers to very simple networks with two anchor nodes shows that the SOCP scales better than $O(k^3)$ and the LP scales better than $O(k^2)$, where k is the number of connections.
More generally, we show that rapid solution of the localization problem for networks with several hundred nodes is possible, and that the technique is directly extensible to networks of thousands of nodes.

Network simulation

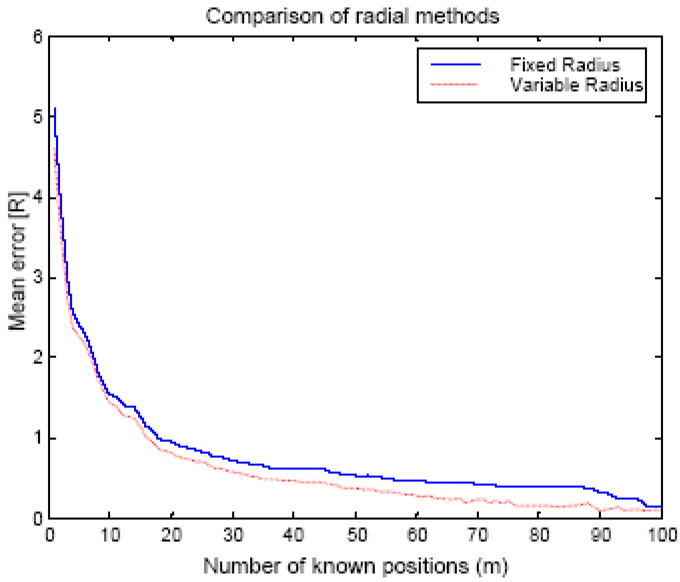
Networks used in the simulations were formed by placing 200 nodes randomly and uniformly in a square region with the side length $10R$. The connectivity is determined by examining pairwise distances; if the distance between two nodes is less than R, the nodes are labeled as connected. Then the largest connected subnetwork of the 200 nodes network is extracted and the node labels are randomly permuted. Ten such networks were used for simulation; the average number of nodes was 194 and the average node connectivity was 5.7.

Comparison between two radial constraint models, comparison with beacon systems

The performance difference between the fixed radius and variable radius RF location methods was measured by performing the following test:

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

Here are the results of those trials:



We also compare these results with a naïve beacon system, where the environment is covered by a grid of anchors. If a node is within the communication range of a beacon, a random guess within this radius of *R* results in the mean error of 2/3 *R* for the network. For our 10*R* x 10*R* network, this performance would require around 50 beacon nodes; this accuracy is achieved with 26 nodes in the variable radius case and 33 for fixed radii with *randomly* chosen known positions.

Significant performance increase with the variable radius method suggests that efforts to enhance distance sensing (either by measuring power directly or by modulating the transmission power through a few discrete steps) will improve position estimation.

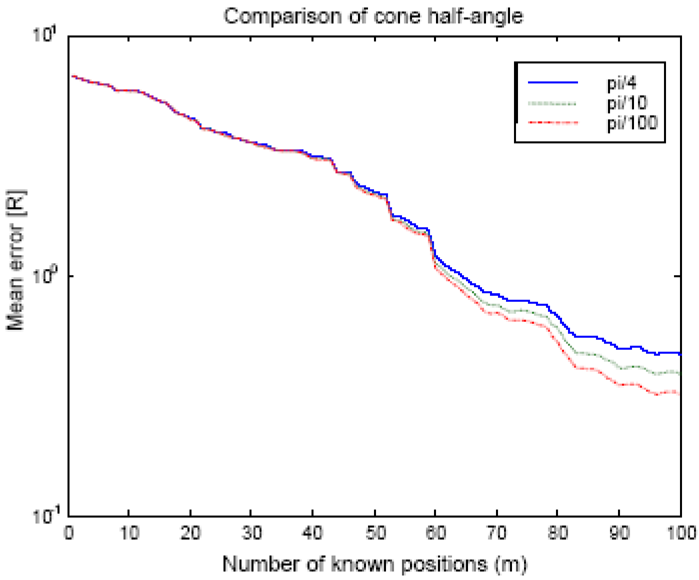
Selection of anchor nodes

Averaged over the 10 test networks, selecting four nodes closest to the corners as anchors reduces the mean error in the variable radius case from 2.4*R* to 1.2*R* (compared with random selection of four anchor nodes). Selecting additional nodes closest to the middle of the external edges for a total of 8 known positions reduces the mean error from 1.7*R* to 0.72*R*. With 8 known positions placed at the network perimeter, the 40+ beacon network performance is matched.
Additionally, selection of the bounding rectangle centers for the unknown positions does improve the estimation accuracy: the mean error drops from 0.72*R* to 0.64*R*.

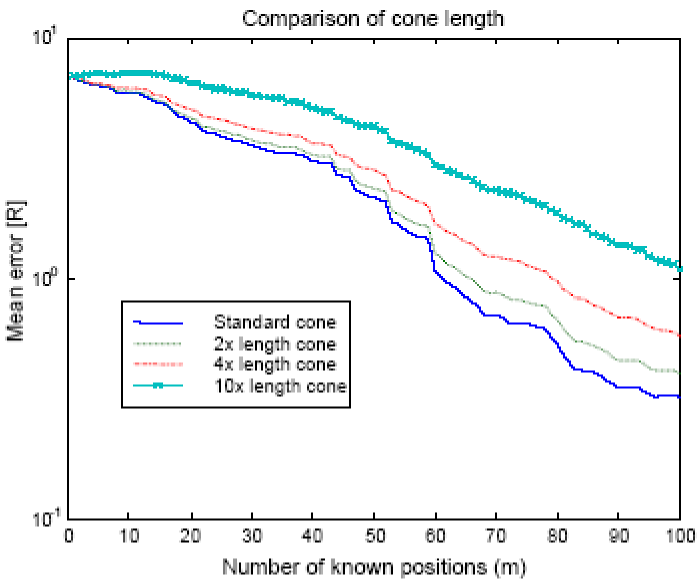
Angular constraint model results

Two parameters were varied in the experiments: the half-angle of uncertainty θ and the distance to the outer bound of the cone.

In the first experiment, θ is reduced from π/4 to π/10 and to π/100. 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:



To determine sensitivity of the results to the uncertainty in the cone length, the outer bound was varied in the second experiment. 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 was varied. A half-angle of $\pi/100$ is used in all the trials.



Finally, we note that the results for the angular and radial methods should not be compared directly as different numerical solvers with different initializations and random objective functions were used.

Dependence on node density and connectivity

The experiments show that increasing graph connectivity improves performance dramatically, but would require significant increase 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.

Application: tracking objects through WSN

A specific application of the techniques described is tracking 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. 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 estimating *k* unknown node positions.

Major deficiencies and research directions

Scalability

As networks grow beyond 2000 nodes, the problems (particularly the radial constraint method) become computationally intensive. This is an alarming result for scaling to networks of hundreds of thousands of nodes. Two options for tackling the issue: limiting the number of constraints at each node and solving the problem hierarchically. In the latter approach we may use clustering algorithms for dividing a networks into a number of subnetworks. Solving the localization problems separately for each subnetwork with respect to its (unknown) center and considering centers as virtual nodes in the larger network, we may achieve good scalability with multiple hierarchical steps.

Anchor nodes placement problem

If the anchor nodes are placed to the interior of the network, the estimated positions will lie inside the convex hull of the anchor nodes, yielding possibly highly inaccurate results.

Erroneous data management problem

The approach does not offer any means for detecting erroneous connections. If a proximity constraint is fallaciously reported, the algorithm will, in general, fail. Testing for such errors is as difficult as solving the position estimation problem itself.

[2] addresses the last two deficiencies by using information contained in non-convex distance constraints.