Distributed Online Localization in Sensor Networks Using a Moving Target

Aram Galstyan[1], Bhaskar Krishnamachari[2],
Kristina Lerman[1], and Sundeep Pattem[2]

[1] Information Sciences Institute
[2] Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, California
{galstyan, lerman}@isi.edu, {bkrishna, pattem}@usc.edu

Abstract. We describe a novel method for node localization in a sensor network where there are a fraction of reference nodes with known locations. For application-specific sensor networks, we argue that it makes sense to treat localization through online distributed learning and integrate it with an application task such as target tracking. We propose distributed online algorithm in which sensor nodes use geometric constraints induced by both radio connectivity and sensing to decrease the uncertainty of their position. The sensing constraints, which are caused by a commonly sensed moving target, are usually tighter than connectivity based constraints and lead to a decrease in average localization error over time. Different sensing models, such as radial binary detection and distance-bound estimation, are considered. First, we demonstrate our approach by studying a simple scenario in which a moving beacon broadcasts its own coordinates to the nodes in its vicinity. We then generalize this to the case when instead of a beacon, there is a moving target with *a-priori* unknown coordinates. The algorithms presented are fully distributed and assume only local information exchange between neighboring nodes. Our results indicate that the proposed method can be used to significantly enhance the accuracy in position estimation, even when the fraction of reference nodes is small. We compare the efficiency of the distributed algorithms to the case when node positions are estimated using centralized (convex) programming. Finally, simulations using the TinyOS-Nido platform are used to study the performance in more realistic scenarios.

1 Introduction

Wireless sensor networks (WSN) hold a promise to "dwarf previous revolutions in the information revolution" [1]. Future WSN are envisioned to consist of hundreds to thousands of sensor nodes communicating over a wireless channel, performing distributed sensing and collaborative data processing tasks for a variety of vital military and civilian applications. Such ubiquitous sensor networks will improve the safety of our buildings and highways, enhance the viability of wildlife habitats, dramatically shorten disaster response times, reduce commute times, and contribute to many other vital functions as part of the "embedded, everywhere" vision.

Node localization is a fundamental problem in sensor networks. Both at the application layers as well as for the underlying routing infrastructure, it is often useful to know the locations of the constituent nodes with high accuracy. There have been a number of recent efforts to develop localization algorithms for wireless sensor networks, most of which are based on using static reference beacons, signal-strength estimation or acoustic ranging. Common characteristics in these efforts have been (i) a view of localization as a one-step process to be performed at deployment time and (ii) the separation of localization from the application tasks being performed.

For application-specific sensor networks, we argue that it makes sense to treat localization as an online distributed problem and integrate it with the application. Our approach exploits additional information gathered by the network over the course of running an application to significantly improve localization performance. The application we consider in this paper is the single-target tracking problem. Target tracking is a canonical application for wireless sensor networks, not only because of its relevance to intelligence gathering and environmental monitoring, but also because it combines the basic challenges of sensor networks: distributed sensing in power constrained, dynamic environments with likely node failures. In order to achieve good accuracy in target localization task, the nodes themselves have to be well localized. Moreover, the localization algorithm should be efficient and scalable. The main contribution of this paper is a simple distributed online scheme for simultaneous target and node localization in networks with a small fraction of reference nodes (*e.g.*, nodes whose exact positions are known *a priori* either through planned placement or through a Global Positioning Systems (GPS) receiver). In the algorithm we propose, sensor nodes use online observations of a moving target to simultaneously improve the estimates of both the target's and their own positions. Each observation adds a geometric constraint on the position of sensor nodes and over time leads to a dramatic improvement in their position estimates. The approach is generalizable to different sensing models.

First we consider a simplified version of the localization task in which a friendly "beacon" broadcasts its coordinates as it moves through the network. We show that this scenario leads to quick convergence to good position estimates. Next, we extend the approach to the more interesting situation in which the target's position is not known *a* priori. In this case, a bootstrapping mechanism provides for the iterative improvement of estimates of both the target's location and that of the sensor nodes. The sensor network is able to track the target more accurately over time by learning better estimates of both node and target positions through sensor observations. In addition to using constraints imposed by target observation, we describe how a node can also exploit information about a target in its vicinity that it *did not observe* to set tighter bounds on its own position. The algorithms we propose are designed to be completely distributed, and require the exchange of information only between neighboring nodes. *To our knowledge, this is the first proposed distributed online localization method that exploits a sensing application to improve its performance over time.*

We undertake a thorough set of numeric experiments to analyze the performance of the proposed localization techniques. The important parameters that we consider include the size of the network, the fraction of known nodes, the radio range and the sensing modality/range. The performance of this learning-based localization scheme is shown as a function of time, showing how the location errors decrease with additional observations. We compare this distributed localization technique with a centralized semi-definite programming which yields a globally optimal solution. Finally, we present results from simulations using the TinyOS-Nido platform to study the perforance in the presence of packet losses and anisotropy in communication and sensing.

2 Constraint-based Localization

Doherty *et al.* [5] introduce an approach to position estimation in a sensor network using convex optimization. If a node can communicate with another node, its position is restricted by the connectivity constraints to be in some region relative to the other nodes. Many such connectivity or proximity constraints define the set of feasible node positions in a network. These constraints can be encoded as a Linear Matrix Inequalities (LMI-s) and solved using convex optimization techniques to obtain position estimates. Doherty *et al.* considered a centralized method where each node relays its connection statistics to a centralized authority which then computes the global solution. However, a centralized approach scales poorly with the size of the network.

Simić and Sastry [7] present a distributed version of the localization algorithm based on connectivity constraints. They consider a discrete network model and derive probabilistic error and complexity bounds. The distributed algorithm has much better scaling properties than a centralized solution and a lower communication cost, because the nodes are not required to relay information; therefore, distributed solutions are more attractive for large networks containing thousands of nodes.

We propose an *online* distributed algorithm in which nodes improve their location estimates by incorporating connectivity constraints as well as constraints imposed by a moving target. Nodes arrive at an initial estimate of their position using connectivity constraints. Nodes then use detection (or non-detection) of a moving target to update their position estimates. Because the sensing range is usually smaller than the communication range, the tighter constraints imposed by the target help localize nodes more accurately. Moreover, repeated observations of a target by different subsets of the network cause the mean network localization error to decrease over time.

3 Distributed Online Localization

Consider an ad hoc wireless sensor network of N nodes randomly deployed in an $L \times L$ square area, communicating over an RF channel. We make the simplifying assumption of a rotationally symmetric communication range whereby each node communicates with neighboring nodes that fall within the disk of radius r centered on the node. The communication range, r, is assumed to be the same for all nodes; therefore, connectivity via RF channel is symmetric. Although the radial communication model may not be a realistic description of wireless sensor networks in physical environments, it is a valid starting point for modeling purposes, and has been studied by various groups in the past. A fraction f of nodes know their positions, for example, because of an on-board GPS device or because they are affixed to known landmarks in the environment.

Let (x_i, y_i) be the actual position of the i^{th} node. For each node the uncertainty in its location is given by a bounding rectangle $Q_i = [x_i^{min}, x_i^{max}] \times [y_i^{min}, y_i^{max}]$, $x_i^{min} \leq x_i \leq x_i^{max}$, $y_i^{min} \leq y_i \leq y_i^{max}$. Initially, we set $Q_i = [0, L] \times [0, L]$. The estimated position of the node is given by the centroid of the bounding box. The error is the difference between the actual and estimated positions.

3.1 Sensing Models

Our approach is general and directly applicable to two different distance sensing models: radial binary detection and distance-bound estimation. In the radial binary detection sensor modal, each sensor can detect whether or not there is a target within range s of the sensor. In the distance-bound estimation model, each sensor can estimate a bound on the distance within which the target must be present. This model implicitly allows the possibility of sensor noise — in particular, if a signal-strength estimate is noisy, then it may be easier to provide an upper bound on the distance estimate. Our approach also works in the special case where exact distance information is available.

3.2 Localization Using a Moving Beacon

First we consider a simple scenario where a moving beacon whose position is known is used to dynamically self-configure a network. As the beacon moves through the network, it broadcasts its coordinates to the nodes which are at most a sensing distance s away from it. Note that since communication between the nodes is not relevant, the problem reduces to studying the behavior of a single node. Every time a node senses the beacon, it generates a new quadratic constraint that it uses to further reduce the uncertainty in its position. This is illustrated in Fig. 1. After the first observation of the beacon (Fig. 1(a)), an unknown node's position is limited to the circle of radius s around the beacon. The second observation of a moving beacon (Fig. 1(b)) further constrains possible node positions, reducing the uncertainty of its position (shaded region). Repeated observations of a moving beacon improve node localization over time.

We approximate the sensing region by a rectangular bounding box, thereby replacing the quadratic constraint by a weaker but simpler linearized constraint (see Fig. 1(b)). Although this approximation overestimates the uncertainty of the node's position, it guarantees that the actual position of the node is always within the bounding box and considerably simplifies the computation of the bounding box.[3] More precisely, let Q be the bounding box for a node. Then, the node will update its bounding box using the constraint imposed by the beacon at position (x_b, y_b) according to the following rule:

[3] Note that we can find a tighter rectangle that bounds the region of intersection of the two circles; however, the extra computational effort does not buy us much better final localization ability.

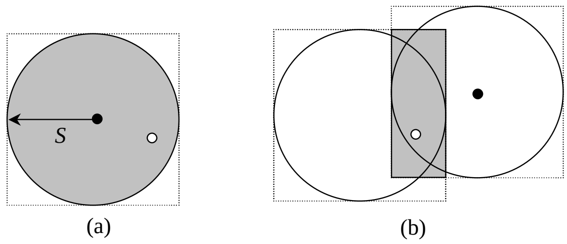


Fig. 1. (a) Sensing constraints limit the set of possible positions of an unknown node (open circle) to the shaded region. Black circle corresponds to a moving beacon with known coordinates. (b) When the node detects a beacon the second time, the uncertainty is collapsed by the observation event. We approximate the sensing regions by squares to simplify computation.

$Q \to Q \bigcap\ [x_b - s, x_b + s] \times [y_b - s, y_b + s]$. As illustrated in Fig. 2, this simple iteration scheme leads to an accurate localization of the node for both models of sensing.

The time-evolution of localization error can be calculated by means of order statistics. Indeed, consider, for instance, the one-dimensional localization problem with binary sensing. Let $\{x_1, x_2, ...x_t\}$ be the set of beacon locations sensed up to time t by a node located at $x = 0$, and let x_{max} and x_{min} me the rightmost and leftmost positions in the set respectively . The localization error $A(t) = 2S - (x_{max} - x_{min})$ is then a random variable with probability density function

$$P(A;t) = \frac{t(t-1)}{(2S)^{t-1}}(2S - A)^{t-2} - \frac{t(t-1)}{(2S)^{t}}(2S - A)^{t-1}, 0 \leq A \leq 2S, \tag{1}$$

and the average localization error goes to zero as $\overline{A(t)} = 4S/(t+1)$. The two-dimensional case can be treated similarly, although the calculation are involved and we do not present the results here due to space limitations. One can instead use simple geometrical arguments to deduce that the asymptotic behavior of the average localization is $\overline{A(t)} \propto 1/t^{2/3}$ for binary sensing, and $\overline{A(t)} \propto 1/t$ for he distance bound model.

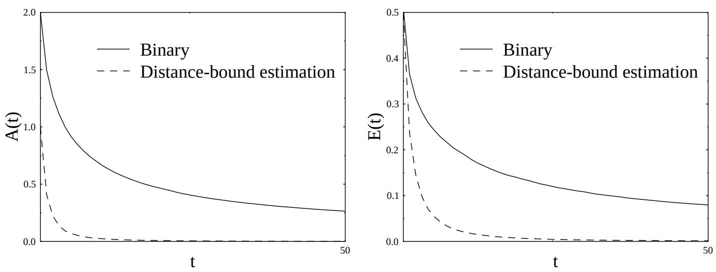


Fig. 2. (a)Average size of the bounding box A and (b) mean square error in node's position E vs time. Average over 500 runs has been taken.

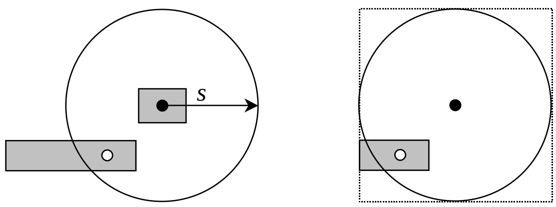
3.3 Localization Using a Moving Target

Now let us assume that instead of a beacon with known coordinates a target with *a-priori* unknown coordinates is observed. The single-node approach of the previous section does not work. However, if there are enough nodes with known positions (or that are relatively well localized) in the vicinity of a given node, then the target can be localized and this information can be used to impose new constraints on the position of the node.

A number of techniques exist for target localization, triangulation being the most popular. However, triangulation requires that the target be observed by three sensors with known locations (in 2D), something which cannot be guaranteed in a mixed sensor network with a small enough f. Here we describe an alternative approach to target localization. Let Q_T be a bounding rectangle for the target (initialized to $[0, L] \times [0, L]$), and let K be the set of nodes that sense the target. Then the bounding box for the target is established as follows:

$$Q_T \rightarrow Q_T \bigcap_{k \in K} [x^k_{min} - S_k, x^k_{max} + S_k] \times [y^k_{min} - S_k, y^k_{max} + S_k], \quad (2)$$

where x^k_{min}, x^k_{max}, y^k_{min} and y^k_{max} give the coordinates of the bounding box for the position of the k^{th} node. For the radial binary sensing model, $S_k = s_k$ is the sensing range (assumed to be the same for all nodes), whereas for the distance estimation model, $S_k = d_k$ is the estimated upper bound on the distance between the k^{th} node and target. Note that this approach guarantees that the actual target position is always inside the bounding box and that the uncertainty region remains convex after the update.



(a) (b)

Fig. 3. (a) Sensing constraints limit the set of possible positions of the sensor node (open circle) to the shaded region. Black circle is a moving target whose position is not known but estimated to be at the center of the shaded region. (b) Observation of the target shrinks uncertainty of the node's position. We approximate the sensing region by a square.

The node can use constraints imposed by the target in two ways. It can use the corners of the bounding box Q_T to impose constraints on its own position (Eq. 3). Alternatively, we can neglect the uncertainty in the target position, and assume that it is located at the center of Q_T (Eq. 4). These update rules are specified below.

$$Q_i \rightarrow Q_i \bigcap [x^T_{min} - S_k, x^T_{max} + S_k] \times [y^T_{min} - S_k, y^T_{max} + S_k], \quad (3)$$

$$Q_i \rightarrow Q_i \bigcap [x^T_{est} - S_i, x^T_{est} + S_i] \times [y^T_{est} - S_i, y^T_{est} + S_i], \quad (4)$$

where x^T_{min}, x^T_{max}, y^T_{min} and y^T_{max} specify the target bounding box, and x^T_{est} and y^T_{est} give the target's estimated position (centroid of the bounding box). S_i is the sensing range for the radial binary sensing model, whereas for the distance-bound model, it is the estimated distance between the i^{th} node and target.

The "centroid" constraint, Eq. 4, is much stronger. We find that despite loosing the guarantee that the unknown nodes remain inside their bounding boxes, for some regimes we can significantly reduce the total network localization error by using this scheme. This is especially useful for the radial binary sensing model

3.4 "Negative" Information

In some cases if a node does not detect a target whereas its neighbor–nodes do, it may be able to use this information to reduce its position uncertainty. This situation is depicted in Fig. 4(a) where, for illustrative purposes, we assume that the target's exact position is known. If the node's actual position was in the shaded region within the circle, it would have detected the target. Hence, the region Q_{neg} can be excluded from the node's bounding box. Note, that we use this "negative" information constraint (i.e. a constraint that is obtained by the act of not sensing a target) only when $Q - Q_{neg}$ is *convex*, where Q is the bounding rectangle for the node. Again, we prefer working with rectangles rather than calculate the exact intersection points. This simplification is shown in Fig. 4(b) where we approximate the circle by an inner square of side $R_{neg} = \frac{2}{\sqrt{2}}S$. Let $Q_{neg} = [x^T - R_{neg}/2, x^T + R_{neg}/2] \times [y^T - R_{neg}/2, y^T + R_{neg}/2]$. Then the updating rule can be written as follows:

$$\begin{aligned} &Q \rightarrow Q - Q \cap Q_{neg} \; if \; Q - Q \cap Q_{neg} \; is \; convex \\ &Q \rightarrow Q \; otherwise \end{aligned} \tag{5}$$

We also studied the case when the circle is approximated by a (larger) square of side $R_{neg} = 2S\gamma$, where $\frac{1}{\sqrt{2}} < \gamma < 1$ (Fig. 4(c)). Although in this case there is a probability the excluded region may in fact contain the node, this approach has an advantage that now the convexity condition on $(Q - Q_{neg})$ is more likely to hold. In fact, as we will show below, the model with $\gamma = 0.9$ produces the best results.

Note also, that if the node keeps track of the "negative" regions for different target positions, these constraints can be combined. Hence, even though each of the constraints individually were not initially useful due to the convexity condition, the combination of both can be used, as illustrated in Fig. 4(d), where the shaded region is the combined Q_{neg}. Although we did not employ this approach in the present paper, we believe that its use can greatly improve the quality of localization.

The generalization of the "negative" constraint rule to the case when the target is within a rectangle $[x^T_{min}, x^T_{max}] \times [y^T_{min}, y^T_{max}]$ is straightforward. The updating rule is exactly the same, and the only difference is the region Q_{neg}, which in this case is constructed as follows: let C_i be a square of side R_{neg} and centered on the $i-th$ corner of the target's bounding rectangle, $i = 1, ..4$. Then Q_{neg} is the intersection of these squares, $Q_{neg} = C_1 \cap C_2 \cap C_3 \cap C_4$. Note that if $x^T_{max} - x^T_{min} > R_{neg}$ or $y^T_{max} - y^T_{min} > R_{neg}$, this intersection will be empty, $Q_{neg} = \emptyset$.

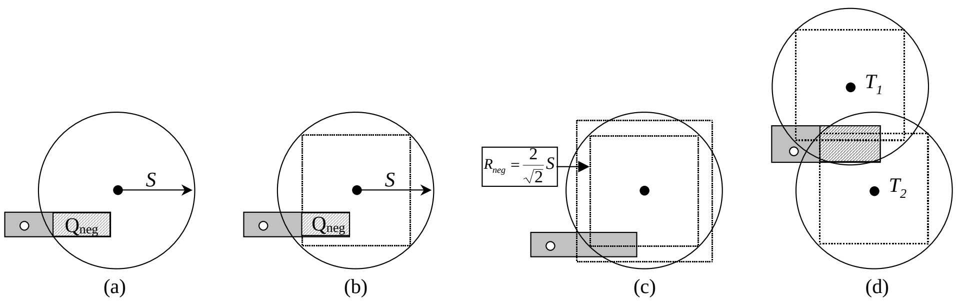


Fig. 4. Different ways in which the node can use "negative" information resulting from the failure to detect a nearby target

3.5 Algorithm

Each node uses an algorithm outlined in Fig. 5 to improve its position estimate. There are three distinct procedures a node uses to update its bounding box Q_i: (i) using constraints imposed by connectivity requirements (Eq. 6), as well as using sensing constraints imposed by the target both when (ii) the node detects a target (Eq. 3–4) and when (iii) no target is detected by a node whose neighbor detects the target (using negative information constraints, see Sec. 3.4). For completeness, we specify the update rule that uses connectivity based constraints:

$$Q_i \rightarrow Q_i \bigcap_{k \in K} [x^k_{min} - r, x^k_{max} + r] \times [y^k_{min} - r, y^k_{max} + r], \tag{6}$$

where x^k_{min}, x^k_{max}, y^k_{min} and y^k_{max} specify the bounding box of node k and r is the radio connectivity range.

```
initialize Q_i = [L, L]
update Q_i using connectivity constraints (Eq. 6)
iterate
    if T is detected
        update Q_T (Eq. 2)
        if T is detected by i
            update Q_i using target information (Eq. 3–4)
        else
            update Q_i using negative information (Eq. 5
        end if
    end if
    for each neighbor k
        if Q_k changes
            update Q_i with connectivity constraints (Eq. 6)
        end if
    end for loop
```

Fig. 5. Pseudocode of the node localization algorithm

4 Results

We carried out extensive numerical simulations of our models described in Sec. 3.3. In all the cases the nodes were distributed randomly and uniformly in an $L \times L$ square area. All the lengths are given in units of L so that we can set $L = 1$. We studied scenarios outlined above for a range of parameters s, r, f and N. For each set of parameters we generated 10–20 different networks, and averaged the results. In all simulations we used only one target that randomly changes its position. Each time a target moves to a new location, τ iterations are performed to allow the constraint imposed by the target to propagate to the nodes. We choose $\tau = 5$ for the results presented here. The total time (counted in number of iterations) of the simulations was $T = 10000$ (hence, 2000 target locations). Of course, real target movements are not random and nodes may be able to learn correlation in target movement and exploit them to arrive at even better position estimates. In this paper, however, we limit the study to the worst case scenario of uncorrelated target movement.

We use two metrics to evaluate the accuracy of localization. Mean error E is the root mean square (rms) of the distance between the nodes actual and estimated (center of the bounding box) positions, averaged over all unknown nodes. The second metric we use is the average size of the bounding rectangle A. Note that these two metrics are related through a constant coefficient for the case when the node's actual position always stays inside the bounding box. This is guaranteed when the nodes use constraints imposed by the bounding box Q_T (though it is not guaranteed when centroid-based constraints are used). We should point out that the existence of such a relationship between average size of the bounding box and position error is useful for online sensor network operation, because the nodes can use the size of the bounding box to estimate their position error at any instant (to our knowledge this connection has not been indicated in any prior work).

4.1 Radial Binary Sensing Model

We assume a homogeneous and uniform system in which all nodes have the same sensing range s. For a given density of nodes, the communication range r was chosen so as to ensure that the network is fully connected. Initially, connectivity based constraints are used to establish bounding boxes for the nodes, then constraints imposed by the target are used to improve node localization.

Figure 6 shows the error E and average size A of the bounding box (averaged over all unknown nodes) versus time for different numbers of the fraction of unknown nodes f. The plot is obtained by averaging results of 20 independent trials to suppress fluctuations. In the results shown, we used $s = 0.12$ and $r = 0.3$. One can see that there is a fast decrease in error, followed by saturation. Note also that the curves shift down as the the number of known nodes increases (lower f). Although we don't show results here, we also found that the localization error decreases with increasing density; this is because at higher node densities, there are more nodes (including known ones) in the vicinity of a target, which helps to better localize the target, and hence, impose stricter constraints on the unknown nodes that sense it.

The results in Fig. 6, which were obtained by using the constraints imposed by the target's bounding box, are rather modest: the average uncertainty in the nodes position can be larger than the sensing range s. The poor performance can be explained by the relatively weak constraints imposed by the bounding box Q_T. This is more pronounced at higher fractions of unknown nodes. However, we can significantly improve node localization results by using constraints imposed by the centroid of Q_T (see Sec. 3.3). In some parameter range, this tradeoff results in an improved performance. Figure 7 shows the average localization error for nodes using the exact (bounding box) approach and using the approximate (centroid) approach as a function of both time and f. Note that the approximate approach reduces the final error by a factor of two as compared to the exact approach.

Figure 8 shows the dependence of the localization error on the communication and sensing range when the nodes use constraints imposed by the bounding box (Fig. 8(a)) and when the nodes use the centroid approach (Fig. 8(b)). The bounding box approach seems to achieve better localization for smaller values of r and s. We believe this is because the localization error is largely determined by the connectivity-imposed constraints, which become less effective as r increases. In the limiting case where each node is connected to all other nodes, the uncertainty in nodes' positions is simply the size of the region L. Therefore, for a high fraction of unknown nodes, the target-imposed constraints are weak due to the initially large uncertainty. For the centroid-based scenario, on the other hand, the situation is different. Even though the uncertainty in the target position can be large, its estimated position will be more accurate as more nodes are sensing it. For larger connectivity range r, there appears to be a value of sensing range that minimizes the rms error.

4.2 Distance-Bound Sensing Model

As we explained above, the poor performance of the binary sensing model is explained by the relatively weak constraints imposed by target detection on the node position. For the distance-bound model, however, the situation is different. Every time a target is at a close distance d from a known node, the uncertainty in the estimated target position will be at most d. Moreover, the constraints imposed by such a target on a given node will be stronger since the distance to the target will be generally smaller than the maximum sensing range s. Hence, we expect the distance-bound sensing model to outperform the radial binary sensing model. In Fig. 9 we plot the average error E versus time for $s = 0.2$, $r = 0.3$, and for various values of the fraction of unknown nodes f. The nodes use constraints imposed by the target's bounding box to update the uncertainty in their own position. After about 100 target observations, the rms localization error is an order of magnitude

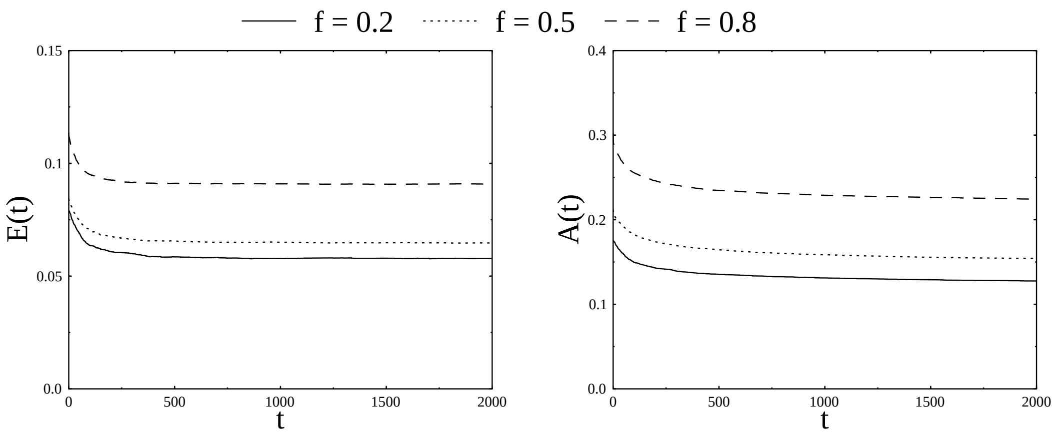


Fig. 6. Mean error E and uncertainty A vs time for the radial binary sensing model

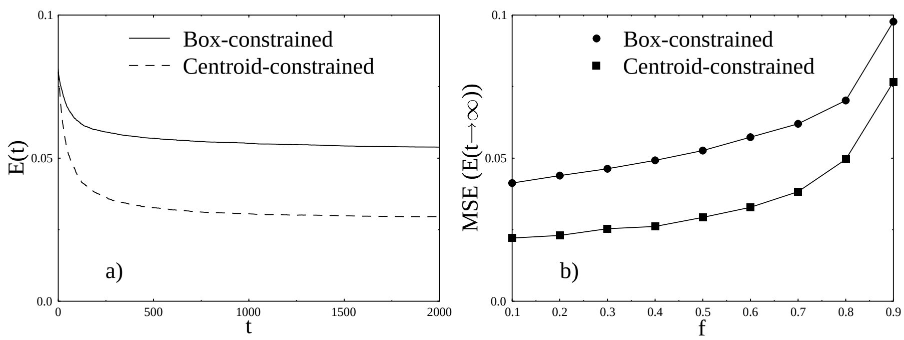


Fig. 7. Comparison of "rigorous" and "approximate" (centroid–based constraints) algorithms a) Time evolution b) Value after saturation

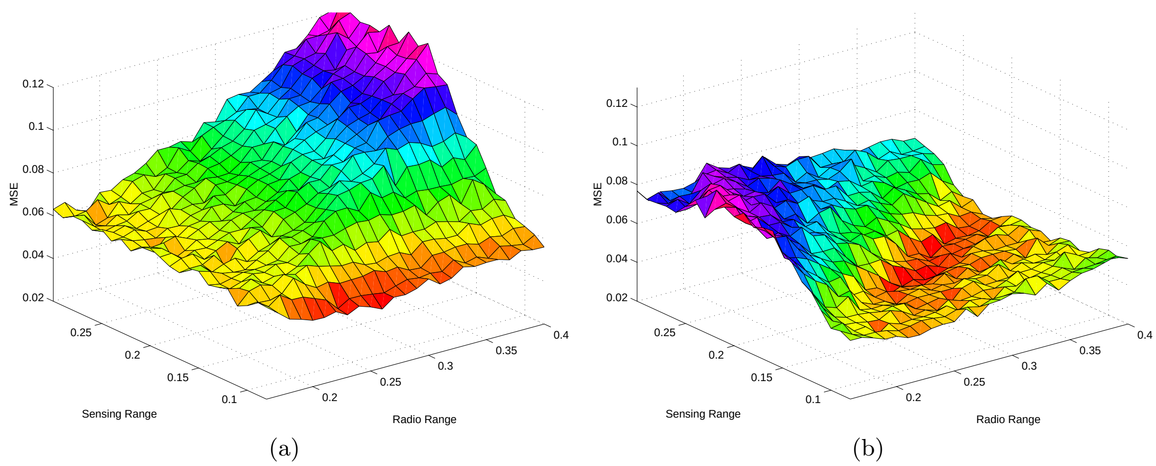


Fig. 8. Dependence of the final rms localization error ($E(t \to \infty)$) on the sensing and radio connectivity ranges using (a) the bounding box approach and (b) using the centroid approach

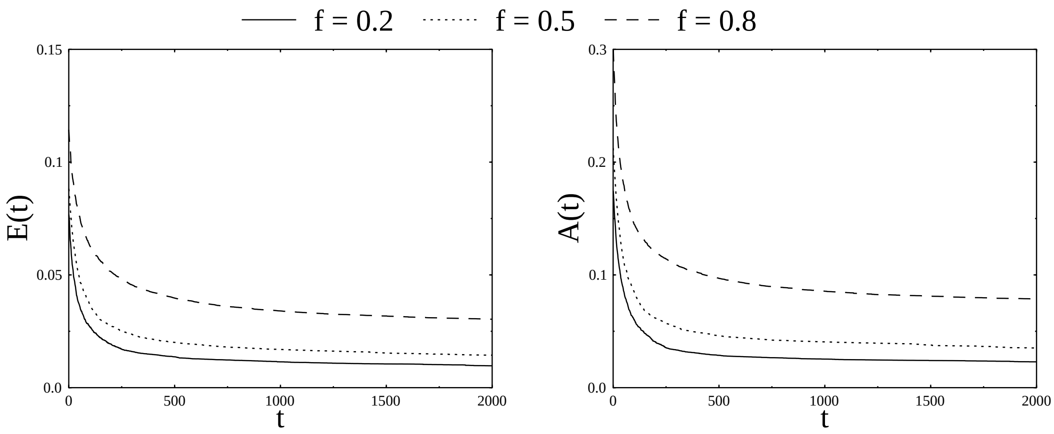


Fig. 9. Error vs time for distance-bound sensing model

smaller than using connectivity-based constraints alone. Remarkably, excellent localization can be achieved even for values of f (fraction of unknown nodes) as high as $f = 0.8$. We attribute this dramatic improvement in performance to the strong constraints imposed by target detection in this sensing model.

Next, we studied the impact of using "negative" information as was described in the Sec. 3.4. Intuitively, its effect is to "push" the nodes away, especially ones that are located at the boundary and tend to have a bounding rectangle biased towards the center. In Fig. 10 we compare the performance of localization algorithm with $\gamma = 0$ ("negative" information is not used), $\gamma = \sqrt{2}/2$ (the inner rectangle is used), and $\gamma = 0.9$. Clearly, introducing "negative" constraints significantly improves the accuracy of localization. Generally speaking, this improvement depends on the other parameters of the model. Note that for $\gamma = 0.9$ the property that the the node is always inside the box does not hold in general; however, the average error (expressed by the metric d) in this case decreases even further compared to the other two "safer" approaches.

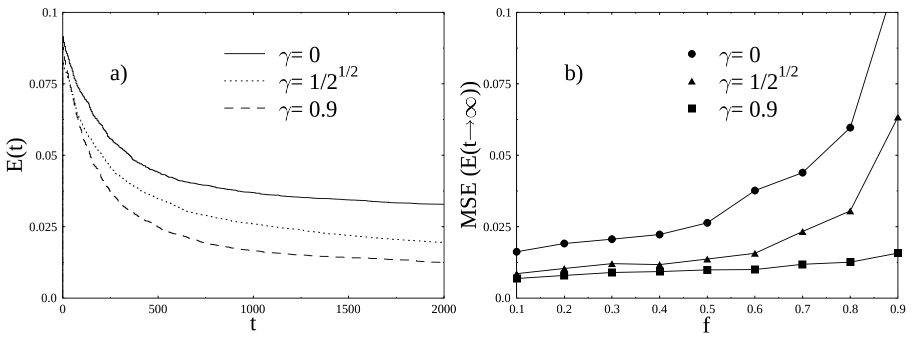


Fig. 10. Effect of the negative constrains on the algorithms a) Time evolution b) The saturated rms error value as a function of f. The parameters are $N = 100$, $r = 0.3$, $s = 0.2$

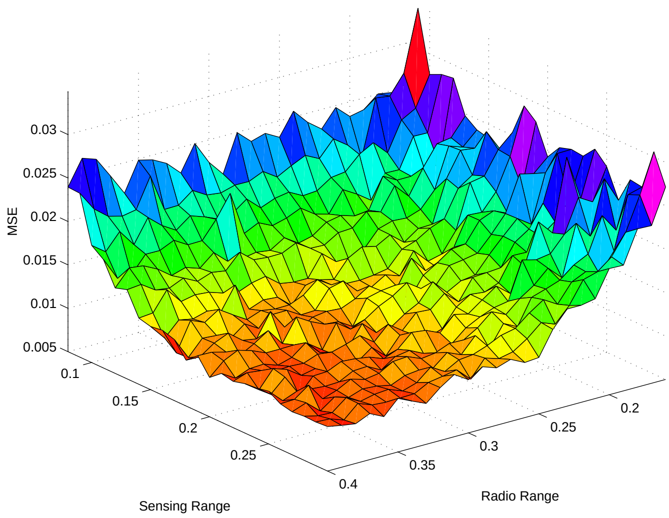


Fig. 11. Mean squared error as a function of r and s for the distance-bound model

In Fig.11 we show the dependence of the localization error on the radio connectivity range r and maximum sensing range s. As one would expect, the best localization is achieved for the larger values of r and s.

Finally, at the end of this section we reflect on how the target localization itself is improved with time. Because of the noise, we had to take an average over for 100 realizations. The result is shown in Fig.11. The error in target localization drops as nodes become better localized themselves and saturates at about a tenth of the maximum sensing range value.

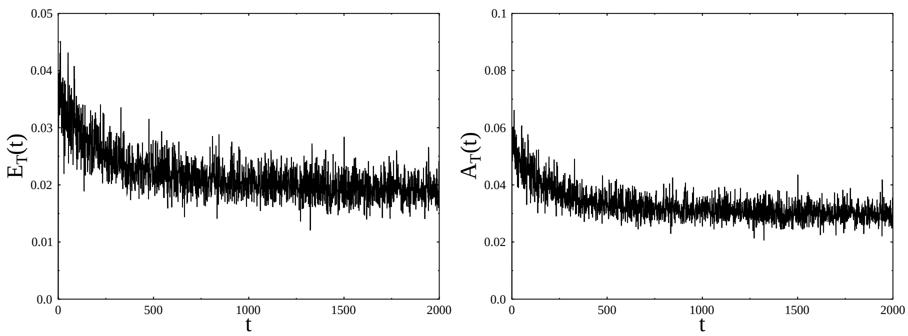


Fig. 12. Time evolution of the error and uncertainty of estimated target position

4.3 Comparison to a Centralized Solution

To estimate the quality of our algorithms, we compared our results with a baseline solution using centralized, semi-definite programming, the approach used for node localization using connectivity based constraints [5]. To account for the constraints imposed by the target, we used an alternative approach to the one considered in the previous sections. Namely, instead of estimating the target's position and then constraining the node, we introduce pair-wise constraints between the nodes that sense the same target. Specifically, let us assume that the target is sensed by i^{th} and k^{th} nodes. Then their positions are constrained via the inequality $|\mathbf{x}_i - \mathbf{x}_k| \leq s_i + s_k$, where $s_i = s_k = s$ for the binary sensing model, and $s_i = d_i$, $s_k = d_k$ for the distance–bound model.

We compared the results of our and centralized algorithms for different values of the fraction of unknown nodes f. The results are presented in Fig. 13. For the binary-detection model, the centralized solution is better than the distributed one for smaller fraction of unknown nodes, and two algorithms perform more or less the same as f is increases. For the distance-bound sensing model, the centralized solution is better when $\gamma = 0$, *i.e.*, when negative constraints are not used. Our distributed algorithm with $\gamma = 0.9$, on the other hand, greatly outperforms the centralized solution.

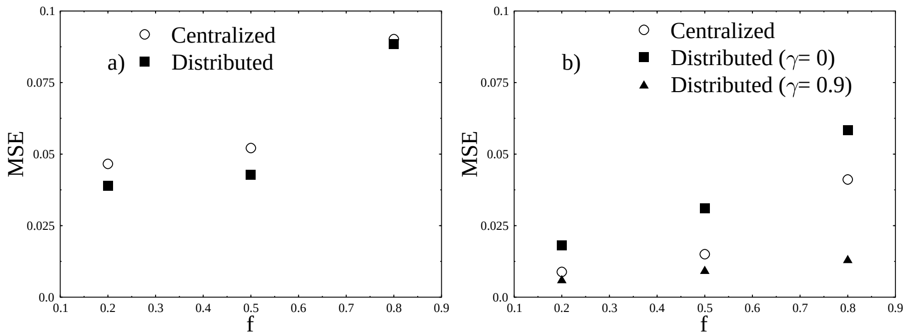


Fig. 13. Performance of the centralized (connectivity constraints only) and distributed localization (including sensing constraints) algorithms (averaged over 10 realizations) for (a) binary-detection and (b) distance-bound sensing models. The parameters are $N = 100$, $r = 0.3$, $s = 0.12$ for binary and $s = 0.2$ for distance bound estimation model

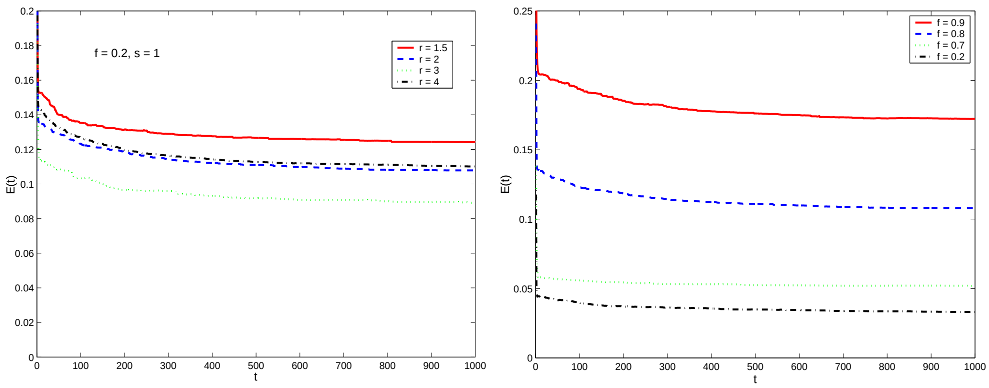
5 Simulations using the TinyOS-Nido platform

In order to validate our algorithms under a more practical setting, we implemented it to work with TinyOS, the operating system used by Berkely motes. We then simulated the performance of the algorithms using the Nido (Tossim) network simulator [11]. Nido works by replacing a small number of TinyOS components and providing additional components like an extensible radio model, ADC and spatial models which together allow users to execute programs that can run on the actual motes. In our simulations, we used one of the Nido nodes as a mobile target. Since Nido cannot simulate a sensing modality, we simulate sensing through communication packets from the target node. To achieve this, a new spatial model was defined in the extensible radio model. In this model, nodes are placed at random in a given 2-D field and connectivity among them is computed given the communication radius. Both the binary and distance bound sensing model algorithms were implemented as follows.

Binary Sensing: At each target position, the target sends out packets with its unique nodeID number. All nodes within a sensing radius receive this packet and communicate to their neighbors that they have sensed the target(using a flag) and also the coordinates of their bounding box. Once a node gets this information from all nodes that have sensed the target, it can compute the target bounding box locally and update its own bounding box.

Distance Bound Sensing: At each target position, the target sends out packets with its exact location and unique nodeID number. All nodes within a sensing radius receive this packet, compute their distance from the target and use this to update the local target bounding box. All neighbors exchange this information and make the final updates to their bounding box as in the binary case.

After all updates, the spatial model allows for changing target node position and recomputing the connectivity matrix. Fig. 14 shows the behavior of the localization error for both models using Nido simulations. It shows that increasing the communication range results in more connectivity constraints but also weaker constraints. Hence, increasing r after a certain value actually results in higher localization error. And we also find here that the performance of the algorithms improves as the fraction f of nodes with unknown positions decreases.



(a) (b)

Fig. 14. Localization error vs time obtained for the distance bound sensing model via Nido network simulations for (a) varying communication ranges and (b) for varying fraction of nodes with known positions .

In order to study the performance of the algorithm in more realistic scenarios, we simulated packet loss due to bit errors and anisotropy in the communication range. Bit errors were introduced by manipulating the transmit function for our new spatial model. Anisotropy in communication range can be modelled by using an additive zero mean Gaussian while computing the connectivity matrix.

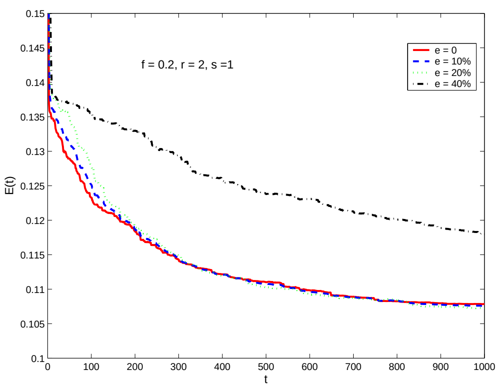
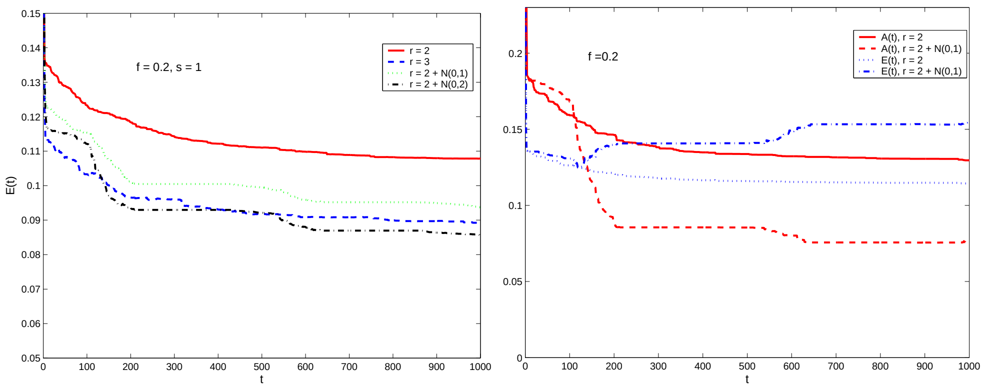


Fig. 15. Localization error vs time obtained via Nido network simulations for varying error rates (distance bound sensing)



(a) (b)

Fig. 16. Localization error E vs time with anisotropy in communication range for (a) distance bound sensing) and (b) binary sensing models (also showing the area of bounding box A) using Nido network simulations.

Since each packet provides a constraint, we can expect that losing packets leads to a slower rate of convergence. Fig. 15 shows the effect of packet loss(e % of all packets are lost) due to bit errors on the distance bound model (the results are similar for the binary sensing model as well). It is observed that the performance is not affected very much with up to 20% packet drops. Fig. 16 (a) shows the effect of anisotropy in communication range for distance bound sensing. Using $r = 2 + N(0, 1)$, some nodes that are separated by distance more than $r = 2$ away are also connected and hence provide stronger constraints(since distance between the nodes is known). In binary sensing, all connected nodes are assumed to be within $r = 2$. It can be seen from Fig. 16 (b) that while the bounding box area A decreases rapidly, the nodes are no longer within the bounding box resulting in an increase in the localization error E.

These results illustrate both the robustness of the proposed algorithms, as well as the feasibility of implementing these lightweight algorithms on a practical sensor network platform such as Motes running TinyOS.

6 Related Work

Providing GPS or precise location information to all nodes may not be feasible in large scale wireless sensor networks due to considerations of cost, or adverse deployment environment (such as indoors or under foliage). A survey of alternative localization techniques for a range of applications is provided in [10]. In particular, prior work for sensor has examined the possibility of providing localization when a few reference or beacon nodes are available [8], [9], [6], [2], [5], [7].

If accurate ranging is available, i.e. the distance to the reference nodes can be measured perfectly through signal-strength based measurements, then multilateration techniques may be used for accurate localization [8], [9], though there may be significant challenges in such an approach when fading and noise are taken into account, as discussed in [6]. Bulusu *et al.* [3] showed that a simpler centroid-based approach can be used with the reference beacons. Doherty *et al.* [5] formalized the localization problem as a convex non-linear program to be solved centrally, using only radio connectivity between the sensor nodes as constraints. Simić and Sastry [7] present a distributed version of the localization algorithm based on connectivity constraints. They consider a discrete network model and derive probabilistic error and complexity bounds.

Savvides, Han and Srivasta in [8] have presented a distributed, iterative technique for multilateration when signal-strength measurements are available. This is improved using a more computationally intensive approach for collaborative multilateration by Savvides, Park and Srivastava in [9].

Common features to these prior efforts is that they separate the localization problem from the application and that the localization is performed in a one-shot manner. Our work significantly extends these prior approaches because our distributed approach permits the nodes in the network to incorporate additional constraints over time through sensor measurements of an unknown target.[4] We exploit the application-specific nature of sensor networks to further optimize for localization.

Paper [4] is closest in spirit to our work, because it too utilizes target tracking to improve node localization using DOA-based techniques. However, there are still significant differences in detail. In [4] it is assumed that the target follows a simple linear trajectory with constant velocity. Further, [4] addresses the localization of nodes in a small-scale sensor array (not a network of sensors) and also use a computationally intensive, centralized extended Kalman filter for the localization, which is not a scalable solution to the problem we consider in this paper.

The related problem of placing reference nodes/beacons adaptively to provide good coverage of the operational region is discussed in [3]. Our work is complementary and can be used with such a beacon placement technique.

7 Discussion

We have described an approach that treats localization in sensor networks as an online learning problem and presented a distributed algorithm for it. One novel aspect of our approach is that we allow nodes to use application-specific information, in this case online observations of a target, to improve estimate of both their own as well as a target's position over time. The nodes can use target information in one of two ways: (i) observation of a target imposes constraints on the node's position, and (ii) if a target is in the vicinity of the node but not detected, the node is also able to use such "negative" information to impose tighter constraints on its own position. The mechanism for using "negative" information is the second novel aspect of our work.

We performed extensive numeric and network level(Nido) simulations of the networks for different number of nodes with known positions, different radio connectivity and sensing ranges, packet drops and anisotropic communication range. Starting from an initial configuration with a small fraction of nodes whose positions are known, the nodes iteratively refined their position estimates, achieving dramatically improved localization for the network as a whole as well as for the target. For the distance-bound sensing model, localization was observed to be an order of magnitude better than using radio connectivity constraints alone, even for relatively large fractions of unknown nodes, while for the binary sensing model we found more modest, though still significant, improvement. "Negative" information also substantially helped on the localization task.

Localization gains depend on the type of constraints being used, especially for the binary sensing model. Interestingly, we found that by sacrificing the guarantee that the target's actual position remain within the bounding box and using constraints imposed by the target's estimated position resulted in significantly better network localization. However, this tradeoff may not always be beneficial, because using the more precise constraints imposed by the bounding box also guarantees a simple relationship between the area of the bounding box and mean position error. Such a relationship is useful for online network operations, where nodes can estimate the (unknown) position error using the (known) area of the bounding box.

A more general sensing model than those considered here is one in which the sensor nodes can measure the distance to a target, using signal strength-based estimation, but the measurement is corrupted by noise. Geometrically, such a sensing model can be represented by an annulus of some radius and width: *i*.e., the target is at least a distance s_{min} away from the node and at most a distance s_{max} from it. The methodology presented here is no longer directly applicable, because

[4] Another way to look at it is to divide the node localization process into two phases - pre-operation and during operation. Existing approaches, such as [6], [2], [5], [7], [8], [9] provide a way for pre-operation localization of a sensor network. Once the sensor network is made operational, our proposed technique could be used to further improve the localization by incorporating constraints from the tracking task during network operation.

the approach no longer guarantees the bounding box remains convex — in fact, observations could split the bounding box into disjoint regions. However, we believe that it is possible to extend our methodology to this more general sensing model.

References

1. D. Estrin *et al. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers*, National Research Council Report, 2001.
2. Nirupama Bulusu, John Heidemann and Deborah Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," *IEEE Personal Communications, Special Issue on Smart Spaces and Environments*, Vol. 7, No. 5, pp. 28-34, October 2000.
3. Nirupama Bulusu, John Heidemann and Deborah Estrin, "Adaptive Beacon Placement," *Proceedings of the Twenty First International Conference on Distributed Computing Systems (ICDCS-21)*, Phoenix, Arizona, April 2001.
4. V. Cevher and J. H. McClellan, "Sensor array calibration via tracking with the extended Kalman filter," *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, pp. 2817-2820, 2001.
5. L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," *Infocom 2001*, Anchorage, AK, 2001.
6. P. Bergamo, G. Mazzini, "Localization in Sensor Networks with Fading and Mobility," *IEEE PIMRC 2002*, Lisbon, Portugal, September 15-18, 2002.
7. "Distributed Localization in Wireless Ad Hoc Networks," *Memo. No. UCB/ERL M02/26*, 2002. Available online at (http://citeseer.nj.nec.com/464015.html).
8. Andreas Savvides, Chih-Chieh Han, Mani B. Srivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," *Mobicom 2001.*
9. A.Savvides, H.Park and M. Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems," *WSNA '02*, 2002.
10. Jeffrey Hightower and Gaetano Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, IEEE Computer Society Press, Aug. 2001.
11. P. Levis *et al.* "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications," *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, November 2003.