(Article begins on next page)

Positioning of Node Using Plane Projection onto Convex Sets

Mohammad Reza Gholami, Mats Rydström, and Erik G. Ström
Chalmers University of Technology
Dept. Signals and Systems
Göteborg, Sweden

***Abstract*—We deal with positioning of node in wireless sensor network (WSN) using received signal strength (RSS) when there is no priori knowledge about path-loss exponent and transmission power. Since the RSS decreases on the average with distance, it carries some information about the distance to an unknown node. By ordering the RSS's, we conclude that there are some convex sets where the position of the unknown node can be found in the intersection of them. We introduce a plane projection onto convex sets (PPOCS) approach to solve the positioning problem. Simulation results show good performance for the new methods compared to other reduced complexity algorithms.**

I. Introduction

Developing advanced technology in designing integrated circuit allows wireless sensor network (WSN) to be used vastly in both civil and military applications. Accurate positioning of nodes is one of the important task which has great effect on the performance of every WSN [1]. Most literatures assumed that there are some anchor nodes, also called reference nodes, which can be used to estimate the position of an unknown node [2], [3].

In general, there are various positioning algorithms based on time of arrival (TOA), time difference of arrival (TDOA), received signal strength (RSS), and angle of arrival (AOA) that can be used in different applications [4]. In this paper, we consider independent RSS measurements collected by anchor nodes. The weighted least squares (WLS) estimation can be employed to find the position of the unknown node when there are some RSS measurements in different nodes [4]. Besides high complexity implementation issue, local minima and saddle points in WLS objective function might decrease the performance of algorithm [5]. To avoid converging to the local minima, a robust positioning algorithm based on projection onto convex sets (POCS) was proposed in [6], [5] which needs a few iterations to converge to global minimum. To define a convex set in this method, an estimation of distance to the unknown node should be initially done based on the RSS measurement in each anchor node. This method of positioning is low complex, and distributes the whole computation over all anchor nodes [6].

In general, positioning algorithms that are based on distance estimation derived from RSS measurements need to know or somehow estimate channel parameters (such as path-loss exponent and transmission power) as well. In practice, however, the transmission power and path-loss exponent might either be unknown or difficult to accurately estimate in each anchor node, hence relaxing the assumption of known channel parameters is quite reasonable. When the path-loss is unknown, there are two different methods of positioning. In the first approach, path-loss exponent needs to be estimated from measurements as well. Some methods of estimation have been proposed in literatures [7], [8], and [9]. In the second method, which is considered in this paper, a model-independent positioning is considered which dose not need the path-loss parameter to estimate the position of the unknown node. In [10] a low complex method was proposed provided a low accurate estimation of an unknown position. Recently, a sequence-based localization (SBL) approach based on partitioning the sensing area into various distinct regions defined by distance rank of anchor nodes to that region was introduced in [11]. To discriminate each region, a unique location sequence is assigned to that area which depends on order of vicinity to anchor nodes. Considering all location sequences in one lookup table and defining sequence corresponding to the unknown node based on the received signal from anchor nodes, the problem of positioning of the unknown node is to find the suitable location sequence in lookup table which is more similar to that one measured in unknown node. Finally, the centroid of the corresponding region is reported as the position of the unknown node. This method even though shows good performance makes a high level of complexity in practical application. It also needs to update its lookup table for moving nodes where their positions change with time. This method is introduced for 2D case, and it seems that generalization to 3D should not be straightforward.

In this paper, we relax assumptions on *a-priori* known path-loss exponent and transmission power to only assume that they are similar for all links, but unknown. By ordering the RSS, we conclude that there are some convex sets which the unknown node can be found in the intersection of them. For 2D space, these convex sets are halfplanes. In order to find the intersection of all convex sets, we introduce a method of plane projection onto convex set (PPOCS) which can be implemented in a fully distributed manner. Simulation results show a reasonable performance for the PPOCS with acceptable complexity.

In [12] the authors consider, among other things, the algorithm we call PPOCS-1. As far as we can tell, their work was not in the open literature at the time we submitted this paper. Hence, it is fair to consider their work as parallel to ours.

This paper is organized as follow: In Sec. II, the problem of positioning is studied with considering signal model. Regarding the RSS, new algorithm based on new geometrical model is introduced in Sec. III. In Sec. IV, various computer simulations are engaged to evaluate the performance of new algorithm.

II. Signal model

Let N anchor nodes with known coordinates $\{\mathbf{z}_i = [x_i, y_i]^T \in \mathbb{R}^2\}_{i=1}^N$ uniformly be located in a region. The ensemble mean power at the i^{th} sensor, measured in dB, can be modeled as [13]

$$P_i = P_0 - 10\beta \log_{10}(\frac{d_i(\mathbf{z}_0)}{d_0}) + w_{P_i} \quad (1)$$

where β is a path-loss factor and P_0 is the received power in dB at calibration distance d_0. w_{P_i} is a log-normal shadowing term, i.e, $w_{P_i} \sim \mathcal{N}(0, \sigma_{Pi}^2)$, which has the significant effect on transmitted signal compared to thermal noise or other fading effects. The RSS is a function of distance $d_i(\mathbf{z}) = \|\mathbf{z} - \mathbf{z}_i\|$, $\|.\|$ denoting Euclidean norm, between anchor nodes and the unknown node, located at coordinates $\mathbf{z}_0 = [x_0, y_0]^T \in \mathbb{R}^2$.
Considering deterministic values for P_0 and β, we can model the random variable P_i as a Gaussian distribution with mean $\mu_i(\mathbf{z}_0) = P_0 - 10\beta \log_{10}(\frac{d_i(\mathbf{z}_0)}{d_0})$ and variance σ_{Pi}^2.
In the positioning algorithms based on distance estimate derived from RSS measurements, the two parameters β and P_0 are normally assumed to be known to the anchor nodes. Assuming known β and P_0, the distance can be estimated using the maximum likelihood estimator as

$$\hat{d}_i(\mathbf{z}_0) = 10^{\frac{(P_0 - Pi)}{10\beta}} \quad (2)$$

In this paper, we suppose deterministic values for β and P_0. The parameter β depends on the environment and P_0 also depends on some parameters like antenna radiation pattern and hardware tolerances, hence they might be unknown in each anchor node. We also assume that there is no correlation between the RSS measurements in different nodes, although when the distances between nodes are decreased, the correlation between the RSS's should be increased.

III. Positioning Algorithm with unknown channel parameters

When there is no knowledge about β and P_0, it is very difficult to obtain information about distance from anchor nodes to the unknown node . Instead of estimating the distance, one can sort the RSS's to find some information about position of the unknown node. In this section, we briefly describe the SBL method and considering the drawbacks of this method, we propose new algorithms based on similar geometry used in the SBL approach. For details of the SBL, see [11].

A. The SBL method

Let us consider a 2D network. For a pairs of anchor node, we draw a perpendicular bisector to the line joining their locations. This line divides the 2D space into two halfplanes which can be uniquely defined based on proximity to anchor nodes. Following for $\frac{N(N-1)}{2}$ unique pairs of anchor nodes, the whole sensing area is divided into some *vertices*, *edges*, and *faces* as shown in Fig. 1 for three anchor nodes. For each region, a unique location sequence is defined which determines the order sequence of anchor nodes' ranks based on its distance from them.

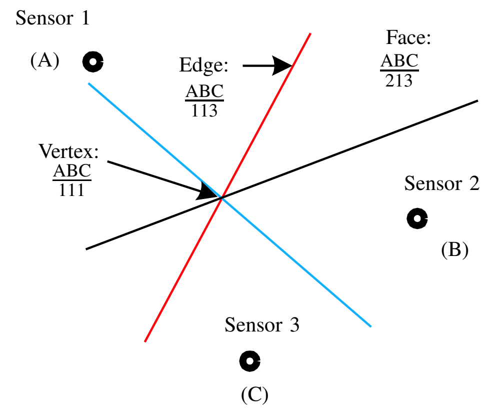

Fig. 1. An example of defining the Edge, Face, and Vertex and their location sequences for 3 anchor nodes

To define the location sequence, a predefined order of anchor node IDs is considered and other location sequences are written based on that. For example, if we consider the predefined order of anchor nodes as ABC in Fig. 1, the location sequence for *Face* is 213 which means anchor node B is the closest node to that area while anchor node C is the farthest node to that area. Similarly, the location sequences for *Edge* and *Vertex* are 113, and 111 respectively.
The SBL algorithm can be implemented as follows [11]:

- Determine all feasible location sequences in the localization space and list them in a location sequence table.
- Determine the location sequence of the unknown node location by using the RSS measurement of localization packets exchanged between itself and the anchor nodes.
- Search in the location sequence table for the "nearest" location sequence to the unknown node location sequence.

The centroid mapped to by that sequence is the location estimate of the unknown node. For a vertex, the centroid is the vertex itself. For an edge, the midpoint of that edge is the centroid and finally for a face the center of the polygon bounding it is the centroid [11](infinite edge or face should be cut down to a finite edge or face).
One drawback of the SBL is the complexity issue due to sequence table construction and searching in that table to find the "nearest" location sequence. The maximum number of *vertexes*, *edges*, and *faces* are $\frac{N^4}{8} - \frac{7N^3}{12} + \frac{7N^2}{8} - \frac{5N}{12}$, $\frac{N^4}{4} - N^3 + \frac{7N^2}{4} - N$, and $\frac{N^4}{8} - \frac{5N^3}{12} + \frac{7N^2}{8} - \frac{7N}{12} + 1$, respectively [11]. Therefore the maximum number of unique location sequences is $\frac{N^4}{2} - 2N^3 + \frac{7N^2}{2} - 2N + 1$ [11].
To determine the "nearest" location sequence, it is needed to compute distance metric between location sequences. Two be-

low metrics between location sequences $U = (u_1, u_2, ..., u_N)$ and $V = (v_1, v_2, ..., v_N)$ are used in SBL [11].

- *Spearman's Rank Order Correlation Coefficient*

$$\rho = 1 - \frac{6\sum_{i=1}^{N}(u_i - v_i)^2}{N(N^2 - 1)} \tag{3}$$

- *Kendall's Tau*

$$\tau = \frac{(n_c - n_d)}{\sqrt{n_c + n_d + n_{tu}}\sqrt{n_c + n_d + n_{tv}}} \tag{4}$$

  Where n_c is the number of concordant pairs, n_d is the number of discordant pairs, n_{tu} is the number of ties in U, and n_{tv} is the number of ties in V. A pairs of location sequence U and V is called concordant if $u_i > u_j \Rightarrow v_i > v_j$ or $u_i < u_j \Rightarrow v_i < v_j$, and called discordant if $u_i > u_j \Rightarrow v_i < v_j$ or $u_i < u_j \Rightarrow v_i > v_j$.

Since the location sequence table is of size $O(N^4)$, searching through it for two metrics Spearman's coefficient and Kendall's Tau takes $O(N^5)$ and $O(N^6)$ operations, respectively. The amount of memory space required is on the order of $O(N^5)$. In Table I the complexity of the SBL is shown versus the number of anchor node N and compared to the other methods [11]. As can seen the SBL has the most complexity in that table comparing to other methods.

TABLE I
COMPLEXITY OF THE SBL [11], 3-CENTROID [11], PROXIMITY [11], AND PPOCS

	SBL	Centroid	Proximity	PPOCS
Time	$O(N^6)$	$O(N\log N)$	$O(N\log N)$	$O(N^2)$
Space	$O(N^5)$	$O(N)$	$O(N)$	$O(N^2)$

B. New Algorithms

As we saw in the previous section, the SBL is a high complex algorithm. In this section, a low complex method of positioning which depends only on the received power is proposed. We assume that both β and P_0 are the same for all links but not known. Like SBL method, we consider 2D network but generalization to 3D is straightforward, but is not considered in this paper.

As mentioned before, the RSS carries some information about distance to unknown node. The ML estimate of the distance to the unknown node from each anchor node, (2) needs some additional information about β and P_0. Instead of solving (2), a simple assumption is considered, namely that the RSS decreases with distance. With this assumption, it is quite reasonable to obtain some information about proximity of the unknown node to the anchor nodes with comparison between RSSs. We can consider the received power in each pairs of anchor nodes and deduce that the unknown node is more likely to be closer to the anchor node with the greatest RSS. Let p_k for $k = 1, 2, ..., \frac{N(N-1)}{2}$ denote all possible anchor node pairs $(\mathbf{z}_i, \mathbf{z}_j)$ such that $1 \leq i < j \leq N$. Given a pair $p_k = (\mathbf{z}_i, \mathbf{z}_j)$, we can divide $\mathbb{R}^2$ into two halfplanes whose boundary is a bisector that is perpendicular to the line that joins the nodes $\mathbf{z}_i$ and $\mathbf{z}_j$. We are primarily interested in the half plane that include the anchor node with the highest RSS measurement. Formally, we define

$$\mathcal{H}_k = \begin{cases} \{\mathbf{z} : (\mathbf{z} - \mathbf{z}_i)^T(\mathbf{z}_j - \mathbf{z}_i) \leq \|\mathbf{z}_j - \mathbf{z}_i\|^2/2\}, & P_i \leq P_j \\ \{\mathbf{z} : (\mathbf{z} - \mathbf{z}_j)^T(\mathbf{z}_j - \mathbf{z}_i) > \|\mathbf{z}_j - \mathbf{z}_i\|^2/2\}, & P_j < P_i \end{cases} \tag{5}$$

From (5), it is obvious that the unknown node most probably can be found in the intersection of $\mathcal{H}_k, k = 1, 2, ..., \frac{N(N-1)}{2}$, i.e,.

$$\hat{\mathbf{z}} \in \mathcal{A} = \bigcap_{k=1}^{N(N-1)/2} \mathcal{H}_k \tag{6}$$

If the intersection $\mathcal{A}$ is empty, due to large-scal fading, the estimator tries to find any point that minimizes the sum of distances to the sets $\mathcal{H}_k, k = 1, 2, ..., \frac{N(N-1)}{2}$,

$$\hat{\mathbf{z}} = \arg\min_{\mathbf{z}} \sum_{i=1}^{N} \|\mathbf{z} - \mathcal{P}_{\mathcal{H}_k}(\mathbf{z})\|^2 \tag{7}$$

Where $\mathcal{P}_{\mathcal{H}_k}(\mathbf{z})$ is the orthogonal projection of $\mathbf{z}$ onto the convex set $\mathcal{H}_k$,

$$\mathcal{P}_{\mathcal{H}_k}(\mathbf{z}) = \arg\min_{\mathbf{w}\in\mathcal{H}_k} \|\mathbf{z} - \mathbf{w}\| \tag{8}$$

Fig. 2 shows the intersection $\mathcal{H}_k$ for three anchor nodes which contains the position of the unknown node.

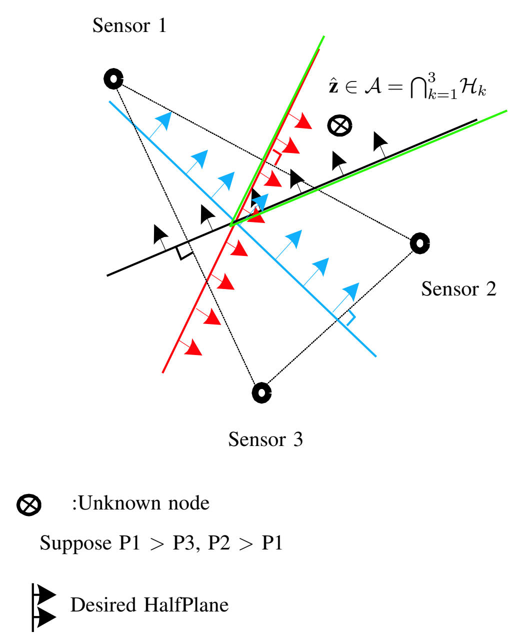

Fig. 2. Intersection of three halfplanes which contains the unknown node

To find the unknown node position in the intersection $\mathcal{A}$, a convex polygonal region [14], we introduce two types of projections onto convex sets to estimate the position of the unknown node.

- ***Algorithm*** 1 (PPOCS-1)
  1. Initialization $\mathbf{z}^{(0)}$ is arbitrary.
  2. For $k = 0$ until convergence or when $k > n\frac{N(N-1)}{2}$ for some n

$$\mathbf{z}^{(k+1)} = \mathcal{P}_{\mathcal{H}_k}(\mathbf{z}^{(k)}) \tag{9}$$

  In algorithm 1, projection is done onto intersection of two halfplanes. It means that algorithm 1 finds a point which has the same distance to each of two anchor nodes, even if they have measured widely different RSSs. Since the primarily assumption about RSS is that its mean decreases with distance, the anchor node with the greater RSS is likely to be closer to the unknown node than is the other node. Considering this fact, we move the boundary of algorithm 1 to inside of halfplane including the anchor node with the greatest RSS.
  We define a halfplane $\mathcal{H}'_k$ similar to $\mathcal{H}_k$, but where the boundary now goes through the node with the strongest received power:

$$\mathcal{H}'_k = \begin{cases} \{\mathbf{z} : (\mathbf{z} - \mathbf{z}_i)^T(\mathbf{z}_j - \mathbf{z}_i) \leq 0\}, & P_i \leq P_j \\ \{\mathbf{z} : (\mathbf{z} - \mathbf{z}_j)^T(\mathbf{z}_j - \mathbf{z}_i) > 0\}, & P_j < P_i \end{cases} \tag{10}$$

- ***Algorithm*** 2 (PPOCS-2)
  POPCS-2 is defined as
  1. Initialization $\mathbf{z}^{(0)}$ is arbitrary.
  2. For $k = 0$ until convergence or when $k > n\frac{N(N-1)}{2}$ for some n

$$\mathbf{z}^{(k+1)} = \begin{cases} \mathbf{z}^{(k)}, & \mathbf{z}^{(k)} \in \mathcal{H}_k \\ (1-\lambda_k)\mathbf{z}^{(k)} + \lambda_k \mathcal{P}_{\mathcal{H}'_k}(\mathbf{z}^{(k)}), & \text{otherwise} \end{cases} \tag{11}$$

  where $\{\lambda_k\}_{k=0}^{\infty}$ are relaxation parameters. In the simulation, the relation parameters are first set to one, and after a given number k_0 of iteration, decreases as

$$\lambda_k = \left\lceil \frac{k - k_0 + 1}{N} \right\rceil^{-1} \tag{12}$$

  where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

Let us consider a finite intersection for all halfplanes, algorithm 1 tries to converge to one point on the perimeter of the intersection area (hence no need to relaxation parameter), while algorithm 2 tries to find a point inside the intersection. In fact, in PPOCS-1, using relaxation parameters might bring a point into the 'wrong' halfplane, which is not desired. Fig. 3 shows the basic concept of two methods of projection.

In the following, we obtain the complexity of the PPOCS algorithms. For every halfplane, algorithm needs to save parameters of a line (the boundary). Every line requires two parameters, and therefore algorithm 1 needs to save $N(N-1)$ variables and algorithm 2 needs to save $2N(N-1)$ variables. Hence PPOCS takes $O(N^2)$ space to save the required variables. We saw through simulation that both algorithms need $n\frac{N(N-1)}{2}$ iterations to converge where $n \ll N$, in most cases $n = 1$ is enough. Hence the complexity of PPOCS is of $O(N^2)$. From Table I, it is clear that the complexity of PPOCS is much lower than SBL.

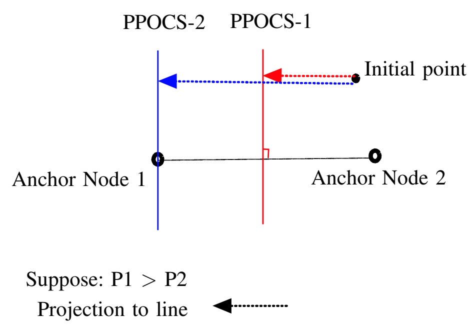

Fig. 3. Projection onto convex set for two types of algorithms

IV. Simulation results

In this section the performance of PPOCS is evaluated through computer simulations. N sensor nodes, uniformly distributed over a square area, were generated. A set of RSS measurements from one unknown node, which was randomly placed in the area, was generated based on the RSS model (1). The cumulative distribution function (CDF) and mean square error (MSE) of the position error $e = \|\hat{\mathbf{z}}_0 - \mathbf{z}_0\|$ were considered to evaluate the performance of various algorithms. In simulation, two methods of PPOCS are evaluated and also the performance of PPOCS is compared to the SBL, Proximity, and 3-centroid [11] algorithms. For more detail about Proximity, and 3-centroid see [11]. For implementation of three mentioned algorithms, we used the C++ code files provided by authors of [11]. It can be downloaded from http://anrg.usc.edu/downloads.html.

In all simulations, the various values of different parameters were set to: $N = 6$, $\sigma^2_{P_i} = 4$ dB, $P_0 = 30$ dB at calibration distance $d_0 = 1$, $\beta = 4$, $\mathbf{z}_i \sim U([-10, 10], [-10, 10])$.

Fig. 4 shows the CDF of different algorithms. As shown, the SBL has the best performance among five algorithms, but, as mentioned before, it has also a high level of complexity. It can also be seen that PPOCS outperforms the Proximity and 3-centroid methods. From Fig. 4, it is clear that PPOCS-2 has better performance rather PPOCS-1.

As mentioned before, PPOCS is an iterative method. In next simulation, we compare the speed of convergence for two version of PPOCS methods. In Fig. 5 the MSE for PPOCS-1 and PPOCS-2 is plotted as a function of the iteration index. It shows that both PPOCS converges after a few iterations. In various simulations, we observed both algorithms converge after $n\frac{N(N-1)}{2}, n < N$ iterations, in most case $n = 1$ is

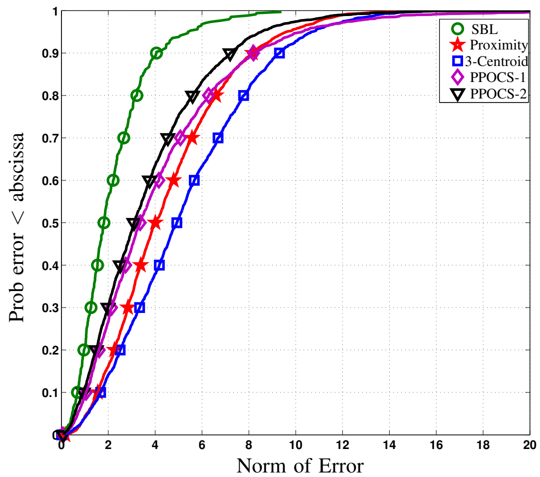

Fig. 4. CDF of $\|\hat{\mathbf{z}} - \mathbf{z}\|$, $\mathbf{z} \sim U([-10, 10], [-10, 10])$, $\sigma^2 = 4$ dB, $\beta = 4, P_0 = 30$ dB

enough. It also shows PPOCS-2 has lower residual MSE compared to PPOCS-1.

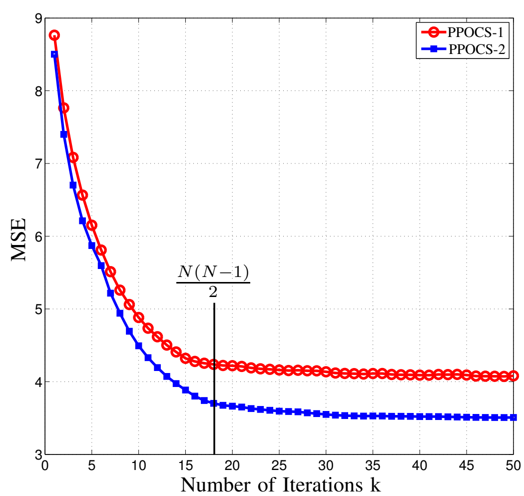

Fig. 5. Convergence rate of PPOCS-1 and PPOCS-2 ($\mathbf{z} \sim U([-10, 10], [-10, 10])$, $\sigma^2 = 4$ dB, $\beta = 4, P_0 = 30$ dB)

V. Conclusions

In this paper, a new method of positioning based on RSS measurements was proposed when path-loss exponent and transmission power are unknown in anchor nodes. Since the RSS on average decreases with distance, we relaxed the assumption of known channel parameters and proposed a method of positioning just by comparing the RSS in each pair of anchor nodes. Defining some convex sets based on the ordering of RSS, we introduced a method of projection onto convex sets to find the position of an unknown node. In this paper, two similar methods of projections with different performance were proposed which distributes the whole process between all pairs of anchor nodes. The complexity of proposed methods is of $O(N^2)$, where N is the number of anchor nodes. For proposed methods, there is a good tradeoff between the performance and complexity compared to other algorithms.

References

[1] R. Huang and G. Zaruba, "Beacon Deployment for Sensor Network Localization," *Proc. IEEE Wireless Communications and Networking Conference*, pp. 3188–3193, March 2007.

[2] A. Sayed, A. Tarighat, and N. Khajehnouri, "Network-Based Wireless Location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[3] C. Chang and A. Sahai, "Cramér-Rao-type Bounds for Localization," *EURASIP Jounal on Applied Signal Processing*, no. 1, pp. 166–166, 2006.

[4] M. Rydström, "Algorithms and Models for Positioning and Scheduling in Wireless Sensor Networks," Ph.D. dissertation, Chalmers University of Technology, 2008.

[5] D. Blatt and A. Hero, "Energy-Based Sensor Network Source Localization via Projection Onto Convex Sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, Sept. 2006.

[6] A. O. Hero and D. Blatt, "Sensor Network Source Localization via Projection Onto Convex Sets (POCS)," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 689–692, Mar. 2005.

[7] S. Mazuelas, A. Bahillo, R. Lorenzo, P. Fernandez, F. Lago, E. Garcia, J. Blas, and E. Abril, "Robust Indoor Positioning Provided by Real-Time RSSI Values in Unmodified WLAN Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, Oct. 2009.

[8] X. Li, "RSS-Based Location Estimation with Unknown Pathloss Model," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3626–3633, December 2006.

[9] J. Shirahama and T. Ohtsuki, "RSS-Based Localization in Environments with Different Path Loss Exponent for Each Link," *IEEE Conference on Vehicular Technology*, pp. 1509–1513, May 2008.

[10] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low-cost Outdoor Localization for very small Devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct 2000.

[11] K. Yedavalli and B. Krishnmachari, "Sequence-Based Localization in Wireless Sensor Networks," *IEEE Transaction on Mobile Computing*, vol. 7, no. 1, pp. 1–14, Jan. 2008.

[12] D. Ampeliotis and K. Berberidis, "Sorted Order-k Voronoi Diagrams for Model-Independent Source Localization in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 426–437, Jan. 2010.

[13] N. Patwari, J. Ash, S. Kyperountas, A. Hero, and N. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Network," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[14] F. P. Preparata and M. I. Shamos, *Computational Geometry*. Springer-Verlag, 1985.