This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

Position Estimation With Moving Beacons in Wireless Sensor Networks

Liang Dong and Frank L. Severance
Department of Electrical and Computer Engineering
Western Michigan University, Kalamazoo, Michigan 49008

***Abstract*— This paper proposes a scheme for position estimation of randomly deployed sensor nodes in a wireless sensor network. Without GPS capability on any of the sensors, the position estimation is facilitated by beacons that move within the network. The beacons are equipped with GPS and can broadcast messages that contain the beacon identifiers and their current positions. With erroneous boundary beacon positions captured at the sensor, the sensor calculates its own position iteratively and updates the estimates upon newly acquired beacon positions. Practical implementation issues are discussed and simulation results show that the proposed iterative approach converges quickly even with beacon positions that have large errors.**

I. Introduction

Position estimation of randomly deployed sensor nodes is an important issue in wireless sensor networks. Accurate positions of sensor nodes improve the routing efficiency and the quality of location-based services [1]. Some positioning schemes require that a number of nodes have GPS capabilities. These positions are propagated through the entire wireless sensor network in order for other nodes to estimate their own absolute positions [2]. However, randomly deployed wireless sensor nodes are usually simple and do not have GPS capability.

A promising scheme for node localization is to use mobile robots in a wireless sensor network [3]. Mobile anchor points are described in [4] to facilitate the localization of sensor nodes. Each anchor is equipped with GPS and has sufficient energy to move in the sensing field and broadcast beacon messages to neighboring sensors. This sensor localization scheme needs no extra hardware or data communication for the sensor nodes. The algorithm can be extended to a more challenging situation in which the beacon's position is not know *a priori* and needs to be estimated through local measurements [5]. A critical implementation concern is that the beacon positions acquired or estimated are usually erroneous [6] or at quantized distance levels [2].

In this paper, we propose a position estimation scheme that uses the assistance of moving beacons in the sensor network. We take into account the reliability of the acquired boundary beacon positions. Positions with large errors are cast. The sensors that receive sufficient boundary beacon positions, at least four non-collinear points, establish confident position estimates through iterative refinements. These position estimates are further propagated in the wireless sensor network for other nodes to compute their own positions. We discuss our work in a two-dimensional wireless sensor network.

Specifically, we evaluate the iterative approach of the sensor position estimation as follows.

1) During the initial positioning process, the sensors that have acquired sufficient boundary beacon positions calculate their own positions as the circle centers and the sensing radii as a least squares solution to a linear equation.
2) When sensors that went through the initial positioning process acquire another group of sufficient boundary beacon positions, they refine the estimates of their positions (circle centers) and the sensing radii through the Levenberg-Marquardt optimization method. The previous estimated circle center positions and radii are used as the initial guesses. The trust region bounds are selected as the standard deviation or the quantization error of the acquired boundary beacon positions.
3) Some restrictions may apply to eliminate beacon position estimates with large errors. These errors can be caused by propagation obstacles in between the transmitting beacon and the receiving sensor.
4) Multiple beacons can be used in the network. For sensors that have never been visited by a moving beacon, their position estimations can be fulfilled by cooperative localization.

II. Sensor Position Estimation with Moving Beacons

As shown in Figure 1, autonomous beacons that are equipped with GPS travel through the sensing field. These beacons periodically broadcast their identifiers and current positions with constant transmit power. Assume that each sensor has an omni-directional receive pattern. That is, the received signal strength (RSS) is related to the distance between the sensor itself and the transmitting beacon, but not to the angle of arrival of the beacon signal. During the positioning process, we define the sensing radius as the largest distance between the sensor and the beacon that the sensor node can correctly detect the beacon message. Note that different sensors may have different sensing radii.

The sensor receives a series of messages from a particular beacon when it moves within the sensing radius. The sensor adds a time stamp to each beacon message it acquires. When the beacon traverses inside the sensing circle, the time stamps are equally spaced. The beginning and the end of this series can be considered, respectively, messages sent when

1525-3511/07/$25.00 ©2007 IEEE

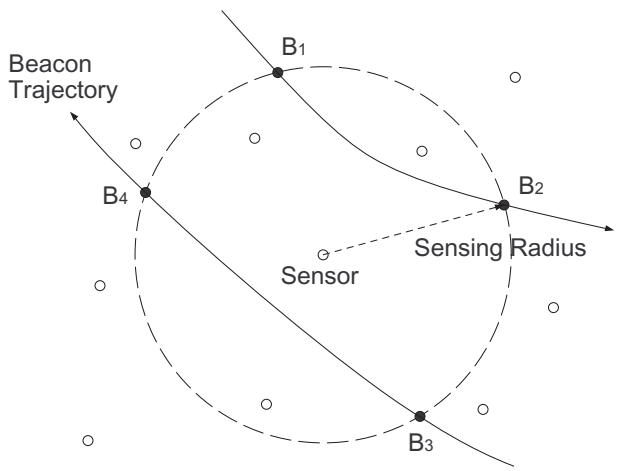

Fig. 1. Sensor positioning with the assistance of moving beacons. Sensor: ○ and Beacon: ●

the beacon enters and leaves the sensing area. We call these two points the boundary beacon points when the beacon is supposedly on the circle whose center is the sensor node and radius the sensing radius. (Pairs $\mathbf{B}_1$ and $\mathbf{B}_2$, $\mathbf{B}_3$ and $\mathbf{B}_4$ are such points in Figure 1.) The sensor collects messages sent at multiple non-collinear boundary beacon points, and uses the acquired boundary positions to compute its own position and sensing radius.

It should be mentioned that we use the time stamp method to simplify the RSS measurement to the connectivity measurement. Connectivity can be considered a binary quantization of the RSS that indicates whether the sensor and the beacon are in communication range [7].

III. Position Estimation Approaches

A. Direct Approach

During the initial positioning process, the sensors that have acquired four or more non-collinear boundary beacon positions calculate their positions and sensing radii directly.

1) Ideal Case – Beacon Points Exactly On A Circle: Given three non-collinear points, $B_1(x_1, y_1)$, $B_2(x_2, y_2)$, and $B_3(x_3, y_3)$, by analysis geometry the center of the circle, $S(c_x, c_y)$, can be found as

$$c_x = \frac{\begin{vmatrix} x_1^2+y_1^2 & y_1 & 1 \\ x_2^2+y_2^2 & y_2 & 1 \\ x_3^2+y_3^2 & y_3 & 1 \end{vmatrix}}{2\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}}, \quad c_y = \frac{\begin{vmatrix} x_1 & x_1^2+y_1^2 & 1 \\ x_2 & x_2^2+y_2^2 & 1 \\ x_3 & x_3^2+y_3^2 & 1 \end{vmatrix}}{2\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}} \quad (1)$$

And the circle radius is given by

$$r = \sqrt{(x_1 - c_x)^2 + (y_1 - c_y)^2} \quad (2)$$

Note that, if the three points are non-collinear, $\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \neq 0$.

2) Non-ideal Case – Erroneous Beacon Points: In practice, the acquired boundary beacon positions may be erroneous. Suppose that M beacon points are detected, that is $\{B_i(x_i, y_i)\}, i = 1, 2, \ldots, M, M > 3$. If these points do not lie on a straight line, the center of the circle, $S(c_x, c_y)$, can be uniquely determined as a least-squares solution to the linear equation

$$\underbrace{2\begin{pmatrix} x_1 - x_2 & y_1 - y_2 \\ x_2 - x_3 & y_2 - y_3 \\ \vdots & \vdots \\ x_{M-1} - x_M & y_{M-1} - y_M \end{pmatrix}}_{\mathbf{A}} \begin{pmatrix} c_x \\ c_y \end{pmatrix} = \underbrace{\begin{pmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 \\ \vdots \\ x_{M-1}^2 - x_M^2 + y_{M-1}^2 - y_M^2 \end{pmatrix}}_{\mathbf{b}} \quad (3)$$

As the M points do not lie on a straight line, $|\mathbf{A}| \neq 0$. We have

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = (\mathbf{A}'\mathbf{A})^{\dagger}\mathbf{A}'\mathbf{b} \quad (4)$$

where † denotes pseudo inverse. And the circle radius is given by

$$r = \sqrt{\sum_{i=1}^{M} ((c_x - x_i)^2 + (c_y - y_i)^2)} \quad (5)$$

B. Iterative Approach

The mobile beacon moves in the sensor network. As soon as a sensor acquires more than three non-collinear boundary beacon positions, it starts to calculate its own position and sensing radius. On one hand, these results may be erroneous due to the inaccuracy of the beacon positions. On the other hand, the beacon continues to move and it may enter the sensing area again. Therefore, the sensor can use the newly acquired boundary beacon positions to refine the estimates of node position and sensing radius.

Suppose that the current estimates of the circle center and radius are $S(c_x, c_y)$ and r, respectively. They are estimated based on the previous M boundary beacon points. Over a period of time, N new boundary beacon points, $B_i(x_i, y_i), i = 1, \ldots, N$, are detected by the same sensor. Now the estimates, $S(c_x, c_y)$ and r, are used as the initial guess of the new search given the new group of N boundary beacon positions. Once the new estimates are derived as $S'(c'_x, c'_y)$ and r', the sensor position and the sensing radius are updated via

$$S(c_x, c_y) := \frac{S(c_x, c_y) \times M + S'(c'_x, c'_y) \times N}{M + N}, \quad (6)$$

$$r := \frac{rM + r'N}{M + N}, \quad (7)$$

and

$$M := M + N. \quad (8)$$

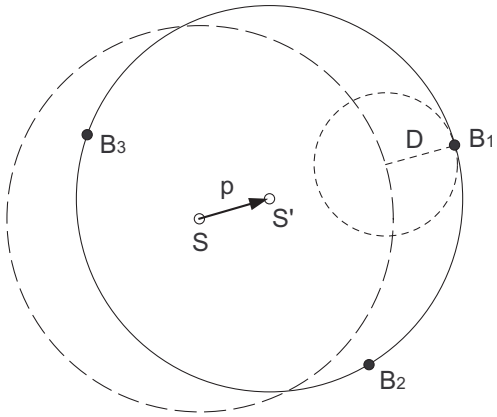

Fig. 2. Trust region bound of the Levenberg-Marquardt method. $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ are the newly acquired boundary beacon positions.

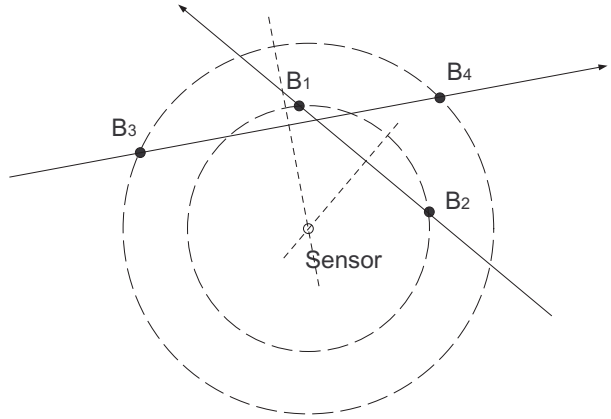

Fig. 3. Concentric sensing circles due to different beacon transmit powers.

To find the new estimates, we construct a cost function that is given by

$$f = \frac{1}{2}\sum_{i=1}^{N}(r_i - r)^2 = \frac{1}{2}\mathbf{F}(\mathbf{c})^T\mathbf{F}(\mathbf{c}) \tag{9}$$

where $r_i = \sqrt{(x_i - c_x)^2 + (y_i - c_y)^2}$, and $\mathbf{F}$ is the vector-valued function given by

$$\mathbf{F}(\mathbf{c}) = \begin{pmatrix} r_1 - r \\ r_2 - r \\ \vdots \\ r_N - r \end{pmatrix} \tag{10}$$

This is a nonlinear least-squares optimum problem that can be written as

$$\min_{\mathbf{c}} f(\mathbf{c}) = \frac{1}{2}\sum_{i=1}^{N} f_i(\mathbf{c})^2 = \frac{1}{2}\mathbf{F}(\mathbf{c})^T\mathbf{F}(\mathbf{c}) \tag{11}$$

where, $\mathbf{c} = (c_x, c_y, r)$, $f_i(\mathbf{c}) = r_i - r$. We could use the Gauss-Newton method to find the minimum of f. But this method is sensitive to the initial guess, and iterations diverge if the guess is too far due to large errors in beacon positions. Therefore, we choose the Levenberg-Marquardt (LM) method [8]. The LM search direction is obtained by solving the linear system

$$\left[\nabla\mathbf{F}(\mathbf{c})^T\nabla\mathbf{F}(\mathbf{c}) + \lambda\mathbf{I}\right] p = -\nabla\mathbf{F}(\mathbf{c})^T\mathbf{F}(\mathbf{c}) \tag{12}$$

where the vector step p is the solution of the minimization problem. ∇ denotes the Jacobian, and $\nabla\mathbf{F}(\mathbf{c})$ is given by

$$\nabla\mathbf{F}(\mathbf{c}) = \begin{pmatrix} \frac{c_x - x_1}{\sqrt{(x_1-c_x)^2+(y_1-c_y)^2}} & \frac{c_y - y_1}{\sqrt{(x_1-c_x)^2+(y_1-c_y)^2}} & -1 \\ \frac{c_x - x_2}{\sqrt{(x_2-c_x)^2+(y_2-c_y)^2}} & \frac{c_y - y_2}{\sqrt{(x_2-c_x)^2+(y_2-c_y)^2}} & -1 \\ \vdots & \vdots & \\ \frac{c_x - x_N}{\sqrt{(x_N-c_x)^2+(y_N-c_y)^2}} & \frac{c_y - y_N}{\sqrt{(x_N-c_x)^2+(y_N-c_y)^2}} & -1 \end{pmatrix}^T.$$

The iteration of estimates is equivalent to calculating the circle centers and radii of groups of N beacon points, and then average them. However, we have a more accurate initial guess for each LM search. Since the LM method is very efficient, this results in less steps for convergence in each search.

The LM method is implemented in the context of a trust-region strategy. The method trusts this model within a limited neighborhood of the point $S(c_x, c_y)$, defined by the constraint

$$||p|| \leq \Delta \tag{13}$$

where, $||\cdot||$ denotes the Euclidean norm. This serves to limit the size of the step taken from S to S'. At the early iterations of the method when S may be far from the circle center, the value of Δ may be small. At later iterations when S is closer to the circle center, it is hoped that there is greater trust in the model. Therefore, Δ can be made sufficiently large.

In our algorithm, because p consists of $\Delta c_x, \Delta c_y$, and Δr, the trust region bound is selected as $\Delta = \sqrt{2}D$. As shown in Figure 2, D can be either the standard deviation σ of the observed boundary beacon positions, or the quantization error of the discretely reported boundary beacon positions from the true boundary circle.

We calculate the Newton step first. That is

$$p = -[\nabla\mathbf{F}(\mathbf{c})^T\nabla\mathbf{F}(\mathbf{c})]^{-1}\nabla\mathbf{F}(\mathbf{c})^T\mathbf{F}(\mathbf{c}) \tag{14}$$

If the Newton direction does not satisfy the constraint $||p|| \leq \Delta = \sqrt{2}D$, we shall determine λ so that

$$\begin{aligned} ||p|| &= ||(\nabla\mathbf{F}(\mathbf{c})^T\nabla\mathbf{F}(\mathbf{c}) + \lambda\mathbf{I})^{-1}\nabla\mathbf{F}(\mathbf{c})^T\mathbf{F}(\mathbf{c})|| \\ &= \Delta = \sqrt{2}D \end{aligned} \tag{15}$$

This is a nonlinear equation in λ. For simplicity, we calculate λ at the first iteration and fix this value for future iterations.

IV. Implementation Issues

A. Multiple Beacons

A single beacon must follow a non-linear trajectory to assist positioning. In the case that it is difficult for the moving beacon to turn sharply and pass through the vicinity of a particular sensor node twice, multiple beacons can be used simultaneously in the network. The broadcasting rates of

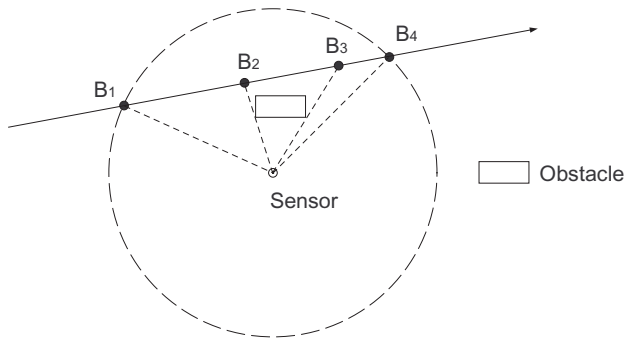

Fig. 4. Propagation obstacle between the sensor and the beacon trajectory.

different beacons can be different to avoid signal collision. If these beacons have the same transmit power, the estimation algorithm remains the same. If beacons have different transmit powers, the sensor can detect pairs of "entering" and "leaving" boundary positions of the same beacon via acquired identifiers. As shown in Figure 3, $\mathbf{B}_1$ and $\mathbf{B}_2$ are such a pair, so are $\mathbf{B}_3$ and $\mathbf{B}_4$. These pairs are on concentric circles, and the sensor position can be estimated by finding the intersection of the perpendicular bisectors of chords [4]. For multiple erroneous beacon position pairs, this can be done by solving an over-determined linear equation.

B. Obstacle Tolerance

There may be propagation obstacles in the wireless sensor network that causes radio irregularity. Suppose that the beacons move in straight lines and the broadcasted messages are too weak to be recognized when it is not in line-of-sight to the sensor. In most cases, the sensor would detect four "boundary" beacon positions as shown in Figure 4. The estimation algorithm ignores the multiple collinear beacon positions with the same beacon identifier. In such a way, most of the obstacle-inflicted erroneous boundary positions can be removed.

Of course the model in Figure 4 does not include all cases. Too many obstacles in the network make the sensing environment unpredictable. The fading effect of the RSS has to be taken into account. The effect of frequency-selective fading can be diminished by using a spread-spectrum method that averages the received power over a wide range of frequencies, and the effect of random shadowing can be modeled as log-normal. A statistical model has to be used to replace the connectivity model with sensing circles. This is beyond the scope of this paper.

C. Cooperative Localization

It may be impractical for the beacons to traverse through the entire network such that they appear within the sensing radius of every sensor node. Multiple beacons traveling in the network can increase the possibility that a particular sensor has multiple visits.

Once a few sensor nodes acquire enough boundary beacon positions and estimate their positions, these estimates can be propagated through the network to assist other nodes to estimate their own positions [2]. The node positions can be estimated given noisy pairwise distance measurements via RSS or time-of-arrival (TOA) [9]. An approach of computing the positions of nodes given only connectivity information is given in [7]. It uses multidimensional scaling technique to transform proximity information into a geometric embedding. A few reference nodes are required to calculate the absolute positions. In a dense wireless sensor network, the proposed beacon-assisted method can locate a small fraction of the sensors. These sparse sensor nodes can be used as reference nodes for cooperative localization for all sensor nodes in the network [1], [10].

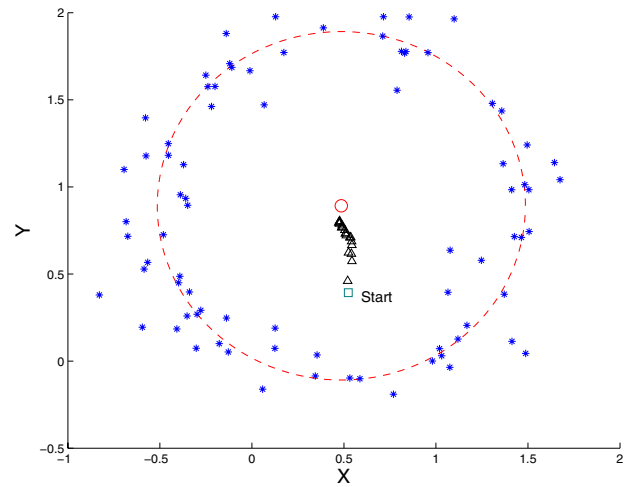

Fig. 5. A single simulation run of the iterative approach to sensor position estimation. $\sigma = 0.2$.

V. Simulations

In this section, we simulate the iterative approach of sensor positioning. Suppose a beacon moves in the vicinity of a particular sensor. The beacon periodically broadcasts its identifier and current position. The sensor keeps a list of the received messages and detects the beacon positions at the sensing boundary. These boundary positions are randomly chosen on the sensing circle with a Gaussian distributed error. Figure 5 shows the exact sensor position and the sensing circle. The $*$ dots indicate the detected boundary beacon positions. Normalized with respect to the sensing radius, the standard deviation of these positions is $\sigma = 0.2$. Suppose the previously estimated sensor position, indicated as a square box in the figure, is erroneous. The previously estimated sensing radius is 0.5, yet the exact sensing radius is 1. When the sensor acquires four new non-collinear boundary beacon positions, it uses the current estimates of sensor position and sensing radius as the initial point to find the new ones using the LM method. Then, the estimates of sensor position and sensing radius are updated according to (6) and (7). This process repeats itself every time the sensor acquires four new non-collinear boundary beacon positions. The iterative estimates of the sensor position are shown as $\triangle$ dots in the figure.

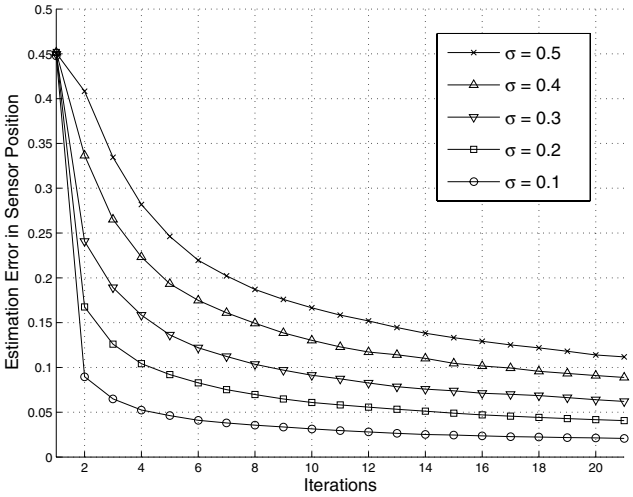

Fig. 6. Estimation error in sensor positions. M = 0.

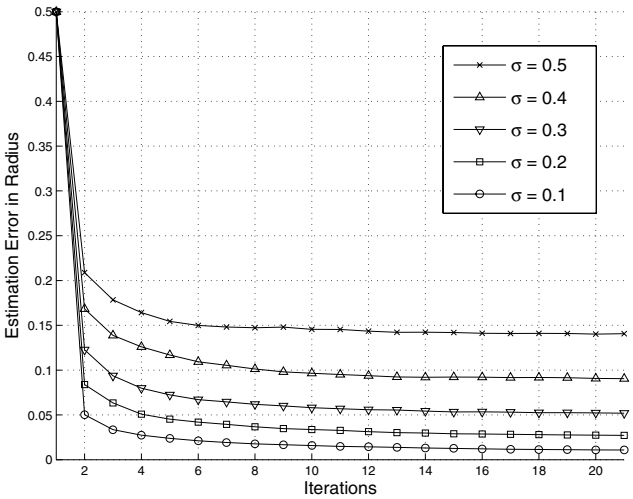

Fig. 7. Estimation error in sensing radius. M = 0.

Figure 6 shows the estimation error in sensor position in the form of $e_c = \sqrt{(\hat{c}_x - c_x)^2 + (\hat{c}_y - c_y)^2}$. Figure 7 shows the estimation error in sensing radius in the form of $e_r = |\hat{r} - r|$. The normalized standard deviation of the boundary beacon positions are $\sigma = 0.5, 0.4, 0.3, 0.2, 0.1$. As the accuracy of the boundary positions increases, the estimation error of the sensor position and sensing radius decreases.

In Figure 6, though we start from an erroneous estimated point of (c_x, c_y, r), we assume no previously acquired boundary beacon positions, i.e. $M = 0$. As comparison, in Figure 8, we assume $M = 20$ previously acquired boundary beacon positions that result in the same estimated point of (c_x, c_y, r). These M positions weight in the update equations (6) and (7). Therefore, Figure 8 demonstrates a slower convergence of the estimates.

VI. Conclusion

An iterative approach to locating wireless sensor nodes with the assistance of moving beacons is proposed. The position estimates are robust against large errors in acquiring the boundary beacon positions. The computational complexity on the sensor nodes is relatively high, however, the acquisition of new boundary beacon positions is continual but infrequent. The Levenberg-Marquardt optimization method is applied and the previously estimated points are used as its initial guess. The algorithm converges to the actual sensor positions quickly. The proposed scheme can be combined with the cooperative localization and be easily implemented in a distributed fashion.

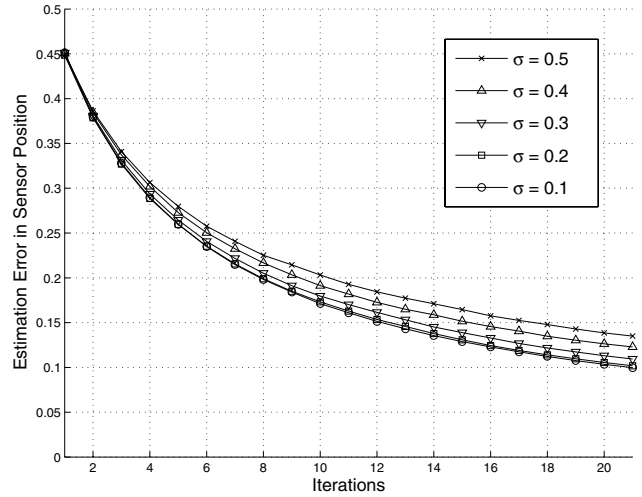

Fig. 8. Estimation error in sensor positions. M = 20.

References

[1] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.

[2] K.-F. S. Wong, I. W. Tsang, V. Cheung, S.-H. G. Chan, and J. T. Kwok, "Position estimation for wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2005, pp. 2772–2776.

[3] P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 3, pp. 285–296, May/June 2005.

[4] K.-F. Ssu, C.-H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 54, no. 3, pp. 1187–1197, May 2005.

[5] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattern, "Distributed online localization in sensor networks using a moving target," in *Third International Symposium on Information Processing in Sensor Networks*, Apr. 2004, pp. 61–70.

[6] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 567–577, Nov/Dec 2005.

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

[8] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*. McGraw-Hill, 1996.

[9] P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Trans. Automat. Sci. Eng.*, vol. 3, no. 4, pp. 360–371, Oct. 2006.

[10] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.