Applied Bionics and Biomechanics 9 (2012) 367–374
DOI 10.3233/ABB-120073
IOS Press

Mobile sensing and simultaneously node localization in wireless sensor networks for human motion tracking

Sen Zhang[a,*], Wendong Xiao[a], Jun Gong[b] and Yixin Yin[a]
[a]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, P.R. China*
[b]*School of Information and Science Engineering Northeastern University, Shenyang, China*

Abstract. This paper exploits optimal position of the mobile sensor to improve the target tracking performance of wireless sensor networks and simultaneously localize both of the static sensor nodes and mobile sensor nodes when tracking the human motion. In our approach, mobile sensors collaborate with static sensors and move optimally to achieve the required detection performance. The accuracy of final tracking result is then improved as the measurements of mobile sensors have higher signal-to-noise ratios after the movement. Specifically, we can simultaneously localize the mobile sensor and static sensors position when localizing the human's position based on augmented extended Kalman filters (EKF). In the algorithm, we develop a sensor movement optimization algorithm that achieves near-optimal system tracking performance. We also presented an sensor nodes management scheme in order to deduce the computation complexity when localizing the static sensor nodes. The effectiveness of our approach is validated by extensive simulations using the simulations.

Keywords: Sensor node localization, mobile sensing, target tracking, augmented EKF

1. Introduction

Human motion tracking in wireless sensor networks is receiving increasing attention from researchers of different fields of study nowadays [1–3]. The interest is motivated by a wide range of applications, such as wireless healthcare, wireless surveillance, human-computer interaction, and so on. In wireless human motion tracking problem, the mobile sensors and static sensors are often applied in one wireless sensor network. In many cases, the sensor nodes' location are unknown in the applications [4, 5]. Because the node localization is a fundamental problem in sensor networks for both the application layers as well as for the underlying routing infrastructure [5], it is often useful to know the locations of the constituent nodes with high accuracy. 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 [7, 8]. Our approach exploits additional information gathered by the network during the course of running a human motion tracking application to significantly improve localization performance.

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 [6, 9, 10]. 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. The application we consider in

*Corresponding author: Sen Zhang, School of Automation and Electrical Engineering, University of Science and Technology Beijing, 30 Xueyuan Road, Haidian District, Beijing 100083, P.R. China, E-mail: zhangsen@ustb.edu.cn.

this paper is the single-target problem to solve: know the current robot location, current human target estimation, and the maximal distance the robot can move omni-directionally, to find the next robot location such that the trace of the target estimation at that location can be minimal. At the same time, how to localize the mobile sensor nodes and static sensor nodes on-line.

Our contributions are threefold: 1) We motivate and propose a novel approach that allows one or more mobile robots to perform node localization in a WSN, eliminating the processing constraints of small devices. Mobility can also be exploited to reduce localization errors and the number of static reference location beacons required to uniquely localize a sensor network. 2) We develop a novel Augmented Extended Kalman Filter (AEKF)-based state estimation algorithm for node localization in WSNs. Localization based on range measurements is solved by treating it as online estimation in a nonlinear dynamic system. Our model incorporates significant uncertainty and measurement errors and is computationally efficient and robust by using the sensor node management scheme proposed in Section 4. 3) Our algorithm is an on-line distributed localization and tracking approach compared to the existing recent work.

2. Mobile sensing with extended Kalman filter

Human motion tracking is receiving increasing attention from researchers of different fields of study nowadays. The interest is motivated by a wide range of applications, such as wireless healthcare, surveillance, human-computer interaction, and so on. A complete model of human consists of both the movements and the shape of the body. Many of the available systems consider the two modelling processes as separate even if they are very close. In our study, the movement of the body is the target. In this section, we proposed how to find the optimal position for the mobile sensor (mobile robot) that can help to obtain the best estimation results when the mobile sensor is tracking a moving target.

We consider the problem of tracking a single human target. Consider the following constant velocity motion model which is used in this paper:

$$X(k+1) = F_k X(k) + w_k \tag{1}$$

with

$$X(k) = \begin{pmatrix} x(k) \\ x_v(k) \\ y(k) \\ y_v(k) \end{pmatrix}, F_k = \begin{pmatrix} 1 & \Delta t_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{pmatrix}, .$$

where $X(k+1)$ is the state of the target at the k-th time step which happens at t_k, $x(k)$, $y(k)$ are x and y coordinates of the target at time step k, $x_v(k)$, $y_v(k)$ are the velocities of the target along x and y directions at time step k, Δt_k is the time interval between the time step k and time step k + 1 which is fixed. $w(k)$ is the Gaussian white acceleration noise with zero mean and covariance matrix Q_k.

$$Q_k = q \begin{bmatrix} \frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2 & 0 & 0 \\ \frac{1}{2}\Delta t_k^2 & \Delta t & 0 & 0 \\ 0 & 0 & \frac{1}{3}\Delta t_k^3 & \frac{1}{2}\Delta t_k^2 \\ 0 & 0 & \frac{1}{2}\Delta t_k^2 & \Delta t \end{bmatrix} \tag{2}$$

Assume the robot can move to take measurement at each time step. The observation model at location $X_s(k) = (x_s(k), y_s(k))$ at time step k is

$$z_{X_s(k)}(k) = h_k(X(k), X_s(k)) + v(k) \tag{3}$$

where $v(k)$ is the Gaussian white measurement noise of the sensor with zero mean and variance R. For example, for robot with ranging sensor, the measurement model is

$$h_k(X(k), X_s(k)) = \sqrt{(x(k) - x_s(k))^2 + (y(k) - y_s(k))^2} \tag{4}$$

EKF operates in the following way [12]: Given the estimate $\hat{X}(k+1|k)$ of $X(k)$, the predicted state $\hat{X}(k+1 \mid k)$ is calculated as

$$\hat{X}(k+1|k) = F_k \hat{X}(k|k) \tag{5}$$

with the prediction error covariance

$$P(k+1|k) = F_k P(k|k) F_k^T + G_k Q G_k^T \tag{6}$$

The predicted measurement for the new robot location at location $X_s(k+1)$ is

$$\hat{z}(k+1|k) = h(\hat{X}(k+1|k), X_s(k+1)) \tag{7}$$

Then the innovation, i.e., the difference between the measurement and the predicted measurement, is given by

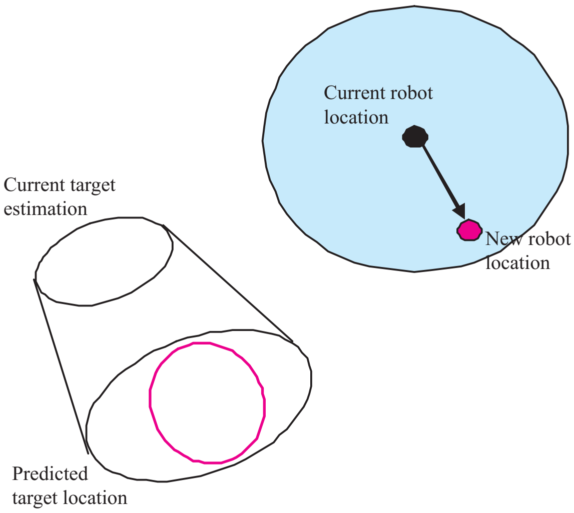

Fig. 1. Target tracking and robot (mobile sensor node) estimation.

$$\gamma(k+1) = z_{s(k+1)}(k+1) - \hat{z}(k+1|k) \quad (8)$$

with the covariance

$$S(k+1) = H_{k+1}P(k+1|k)H_{k+1}^{T} + R(k+1) \quad (9)$$

where $H(k+1)$ is the Jacobian matrix of the observation function h_k with respect to the predicted state $\hat{X}(k+1 \mid k)$. The EKF gain is given by

$$K(k+1) = P(k+1|k)H_{k+1}(\hat{x}(k+1|k), x_s(k+1))S^{-1}(k+1) \quad (10)$$

and the state will be updated as

$$\hat{X}(k+1|k+1) = \hat{x}(k+1|k) + K(k+1)\gamma(k+1) \quad (11)$$

with the error covariance matrix

$$P(k+1|k+1) = P(k+1|k) - K(k+1) S(k+1)K^{T}(k+1) \quad (12)$$

Or equivalently using the information filter, we have

$$P(k+1|k+1) = (P(k+1|k)^{-1} + H_{k+1}^{T}(\hat{X}(k+1|k), X_s(k+1))R(k+1) H_{k+1}(\hat{X}(k+1|k), X_s(k+1)))^{-1} \quad (13)$$

For ranging sensor,

$$H_{k+1}(\hat{X}(k+1|k), X_s(k+1)) = \begin{Bmatrix} \frac{\hat{x}(k+1|k)-x_s(k+1)}{\sqrt{(\hat{x}(k+1|k)-x_s(k+1))^2+(\hat{y}(k+1|k)-y_s(k+1))^2}} \\ 0 \\ \frac{\hat{y}(k+1|k)-y_s(k+1)}{\sqrt{(\hat{x}(k+1|k)-x_s(k+1))^2+(\hat{y}(k+1|k)-y_s(k+1))^2}} \\ 0 \end{Bmatrix}^{T} \quad (14)$$

which is a nonlinear function of new robot location $X_s(k+1) = (x_s(k+1), y_s(k+1))$. As shown in Fig. 1, the mobile sensing problem is to find the best $X_s(k+1)$ according to the following optimization problem (suppose to based on the trace of the covariance matrix):

$$Min\ \ trace\ \ (P(k+1|k+1) \quad (15)$$

under the constraints $\| X_s(k+1) - X_s(k) \| \leq L$

where L is the maximal moving distance of the robot. See Fig. 1.

It's a constrained optimization problem and can be solved by some nonlinear optimization approach. Here we will apply downhill simplex method.

$$Min\ \ Trace(P(K+1|K+1)) + \lambda(L - \| X_s(k+1) - X_s(k) \|) \quad (16)$$

2.1. *The optimization algorithm–downhill simplex*

To solve equation (16), we have to use nonlinear optimization algorithm. The **downhill simplex method** or **amoeba method** is a commonly used nonlinear optimization algorithm. It is due to Nelder & Mead (1965) and is a numerical method for minimizing an objective function in a many-dimensional space [11].

The method uses the concept of a simplex, which is a polytype of $N+1$ vertices in N dimensions; a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so forth.

Like all general purpose multidimensional optimization algorithms, Nelder-Mead occasionally gets stuck in a rut. The standard approach to handle this is to restart the algorithm with a new simplex starting at the current best value. This can be extended in a similar way to simulated annealing to escape small local minima.

Many variations exist depending on the actual nature of problem being solved. The most common, perhaps, is to use a constant size small simplex that climbs local gradients to local maxima. Visualize a small triangle on an elevation map flip flopping its way up a hill to a local peak. This, however, tends to perform poorly against the method described in this paper because it makes small, unnecessary steps in areas of little interest.

The NM algorithm details are as follows [11]:

Assume the objective function that will be maximized is $f(x)$ and x is the variable.

1. order according to the values at the vertices:

$$f(x_1) \leq f(x_2) \leq f(x_3) \leq \ldots \leq f(x_{n+1}) \tag{17}$$

2. compute a **reflection**:

$$x_r = x_o + \alpha(x_o - x_{n+1}) \tag{18}$$

x_o is the center of gravity of all points except x_{n+1}.

If

$$f(x_1) < f(x_r) < f(x_n) \tag{19}$$

then we compute a new simplex with x_r and by rejecting x_{n+1}. Go to step 1.

3. expansion: If

$$f(x_r) < f(x_1) \tag{20}$$

then calculate

$$x_e = \rho x_r + (1 - \rho)(x_o \tag{21}$$

If
$$f(x_e) < f(x_r) \tag{22}$$

compute new simplex with x_e and go to Step 1. Else compute new simplex with x_r and go to Step 1.

4. contraction: If $f(x_n) \leq f(x_r)$ let $x_c = x_{n+1} + \gamma(x_o - x_{n+1})$, if $f(x_c) < f(x_r)$

compute new simplex with x_c. Go to Step 1. Else go to Step 5.

5. shrink step: Compute the n vertices evaluations:

$$x_i = x_1 + \sigma(x_i - x_1) \tag{23}$$

for all $i \in 2, \ldots, n + 1$ go to Step 1.

It is noted that α, ρ, γ and σ are respectively the reflection, the expansion, the contraction and the shrink coefficient. Standard value are $\alpha = 1$, $\rho = 2$, $\gamma = 0.5$ and $\sigma = 0.5$.

For the **reflection**, since x_{n+1} is the vertex with the higher associated value along the vertices, we can expect to find a lower value at the reflection of x_{n+1} in the opposite face formed by all vertices point x_i except x_{n+1}.

For the **expansion**, if the reflection point x_r is the new minimum along the vertices we can expect to find interesting values along the direction from x_o to x_r.

Concerning the **contraction**: If $f(x_r) > f(x_n)$, we can expect that a better value will be inside the simplex formed by all the vertices x_i.

3. Simultaneously Static Sensor Nodes Localization (SSSNL)

In this paper, we will not only decide the optimal position of the mobile sensor/robot at each time step when we track the target, but also we want to simultaneously localize the static sensor nodes and mobile nodes. In order to do this, we use the Augmented Extended Kalman Filter (AEKF) to simultaneously localize the static sensor nodes and mobile sensor nodes when we track the target. Assume the position of the ith static sensor node is denoted by p_i. The system state equation for the ith static sensor node is

$$p_i(k+1) = p_i(k)$$

The static sensor nodes are assumed to be stationary all the time and the number of static sensor nodes in the environment is assumed to be N. Thus the augmented state equation of the mobile sensor (we assume there are only one mobile sensor in the environment) and all static sensor nodes are expressed as follows:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{v}(k) \tag{24}$$

where $\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}_v(k) \\ p_1(k) \\ \vdots \\ p_N(k) \end{bmatrix}$, and $\mathbf{x}_v(k)$ is the mobile sensor node position; $\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} \mathbf{f}_v(\mathbf{x}_v(k), \mathbf{u}(k)) \\ p_1(k) \\ \vdots \\ p_N(k) \end{bmatrix}$,

$$\mathbf{v}(k) = \begin{bmatrix} \mathbf{v}_v(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The static sensor nodes and mobile sensor can get observations of the relative positions between static sensor nodes and the mobile sensor. The observation model of the ith static sensor node is expressed as follows:

$$\mathbf{z}_i(k) = \mathbf{H}_i(\mathbf{x}_v(k), p_i(k)) + \mathbf{w}_i(k) \tag{25}$$

where $\mathbf{w}_i(k)$ is a white noise with zero mean and variance σ_r. The observation function $\mathbf{H}_i(\cdot,\cdot)$ gives the relationship between the sensor measurement and the system state variable when observing the ith static sensor node. We apply the EKF for the state estimation of the mobile sensor and static sensor node. Given the estimate $\hat{\mathbf{x}}(k \mid k)$ of $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$, the predicted state $\hat{\mathbf{x}}(k+1 \mid k)$ using (24) is given by

$$\hat{\mathbf{x}}(k+1 \mid k) = \mathbf{f}(\hat{\mathbf{x}}(k \mid k), \mathbf{u}(k)). \tag{26}$$

The prediction error covariance is approximately given by:

$$\mathbf{P}(k+1 \mid k) = \mathbf{F}(k)\mathbf{P}(k \mid k)\mathbf{F}^T(k) + \mathbf{Q}(k) \tag{27}$$

where $\mathbf{F}(k)$ is the transition matrix of Equation (24) after the linearization. $\mathbf{P}(k \mid k)$ is the prior error covariance estimation at time k. $\mathbf{Q}(k)$ is the covariance of the white noise $\mathbf{v}(k)$, i.e. $\mathbf{Q}(k) = \mathrm{diag}\{\mathbf{Q}_v(k), 0, \cdots, 0\}$.

In view of (25), the predicted measurement is simply

$$\hat{\mathbf{z}}_i(k+1) = \mathbf{H}_i(\hat{\mathbf{x}}_v(k+1 \mid k), \hat{p}_i(k+1 \mid k)) \tag{28}$$

where $\hat{\mathbf{x}}_v(k+1 \mid k)$ and $\hat{p}_i(k+1 \mid k)$ are the elements of $\hat{\mathbf{x}}(k+1 \mid k)$ which is calculated from Equation (26). Then, the difference between the measurement and the predicted observation, namely the innovation, is given by

$$\nu_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1). \tag{29}$$

Thus, the covariance of the innovation is:

$$\mathbf{s}_i(k+1) = \nabla\mathbf{H}_i(k+1)\mathbf{P}(k+1 \mid k)\nabla\mathbf{H}_i(k+1)^T + \sigma_r^2 \tag{30}$$

where $\nabla\mathbf{H}_i(k+1)$ is the Jacobian matrix of the observation function with respect to the predicted system state $\hat{\mathbf{x}}(k+1 \mid k)$. Because each observation is only a function of the sensor node being observed, the matrix is a sparse matrix of the form:

$$\nabla\mathbf{H}_i(k+1) = [\nabla_v\mathbf{H}_i(k+1)\ \ \mathbf{0}\ \ \ldots\ \ \mathbf{0}\ \ \nabla_\mathbf{i}\mathbf{H}_\mathbf{i}(\mathbf{k}+\mathbf{1})\ \ \mathbf{0}\ \ \ldots] \tag{31}$$

where $\nabla_v\mathbf{H}_i(k+1)$ and $\nabla_i\mathbf{H}_i(k+1)$ are the Jacobians of the observation function with respect to the mobile sensor states and the ith static sensor node states, respectively. The EKF gain is given by

$$\mathbf{K}_i(k+1) = \mathbf{P}(k+1 \mid k)\nabla\mathbf{H}_i(k+1)^T\mathbf{s}_i^{-1}(k+1). \tag{32}$$

At time $k+1$, we use new matched observations one by one in the current sensor information (that means the sensor readings that the mobile sensor node received from the static sensors close to it) to update the estimate using the following equations:

$$\hat{\mathbf{x}}_1^- = \hat{\mathbf{x}}(k+1 \mid k), \tag{33}$$

$$\hat{\mathbf{P}}_1^- = \hat{\mathbf{P}}(k+1 \mid k), \tag{34}$$

$$\hat{\mathbf{x}}_i^+ = \hat{\mathbf{x}}_i^- + \mathbf{K}_i(k+1)\nu_i(k+1), \tag{35}$$

$$\hat{\mathbf{P}}_i^+ = \hat{\mathbf{P}}_i^- - \mathbf{K}_i(k+1)\mathbf{s}_i(k+1)\mathbf{K}_i^T(k+1), \tag{36}$$

$$\hat{\mathbf{x}}_{i+1}^- = \hat{\mathbf{x}}_i^+ \quad (i = 1, \cdots, n), \tag{37}$$

$$\hat{\mathbf{P}}_{i+1}^- = \hat{\mathbf{P}}_i^+ \quad (i = 1, \cdots, n), \tag{38}$$

$$\hat{\mathbf{x}}(k+1 \mid k+1) = \hat{\mathbf{x}}_n^+, \tag{39}$$

$$\mathbf{P}(k+1 \mid k+1) = \hat{\mathbf{P}}_n^+ \tag{40}$$

where i means the ith observation from static sensor node i and n is the number of static sensor nodes that the mobile sensor can hear from in the current time step; $\hat{\mathbf{x}}_i^-$ is the system state estimate before update using the ith observation and $\hat{\mathbf{x}}_i^+$ is the estimate after the update by observation i. $\hat{\mathbf{P}}_i^-$ and $\hat{\mathbf{P}}_i^+$ are the corresponding state covariance matrices, respectively.

4. Sensor nodes localization management scheme

If the static sensor nodes' location estimation is to be built incrementally as information is gathered from sensors, there is typically a need for a sensor node localization management process in order to prevent the heavy computational burden when the system state matrix is augmented. This process has the function of managing the information present in the knowledge base and possibly aiding the sensing process. Given the fact that computational resources are limited, an information management technique that reduces the stored data without sacrificing much information is required. To improve the applicability of a spatial description to a larger variety of scenarios, it should present the ability to iteratively adapt its geometry to application-specific requirements. The sensor node management process can be divided into three aspects for SSSNL in dynamic environments as follows:

1. Adding Observed Sensor Nodes. When a sensor node observed in the current scan cannot be matched to the existing sensor node list, a new sensor node is initialized.

2. Removing redundant sensor nodes. If all static sensor nodes are included for updating the state, the computational requirement will be high. Thus, redundant sensor nodes that have not been observed for a long time interval should be removed.

3. Removing unstable sensor nodes. sensor nodes become unstable or obsolete if they move or become permanently occluded. For example, sensor nodes might be stationary for a long period of time, and can be considered suitable sensor nodes for SSSNL. But if they move, they are unstable sensor nodes and should be removed from the sensor management scheme. Another case is that structural changes may occur in the environment–such as some static sensor nodes removed. Other cases, such as, an object might be placed in front of a sensor node, occluding it from view. For whatever reason, some sensor nodes may cease to exist and no longer provide useful information. These unstable sensor nodes should be deleted from the sensor management scheme.

After data association, if a sensor node cannot be matched to any existing sensor node in the map, it is considered as a new sensor node. The sensor node initialization is activated. Otherwise, this observation is used for the system update.

After a specified time interval, we shall check if this sensor node is still matched by any new coming observations during this period. If it is matched by none of the observations sensed from external sensors within the specified interval, this sensor node should be removed from the sensor node listing. Otherwise, this sensor node will still be kept in our system variables.

5. Simulation results

This section will present the simulation results. In the first simulation, we apply the mobile robot as the mobile sensor node and 8 static sensor nodes are also employed for the target tracking. We compared the tracking results in Fig. 2 when a mobile sensor node is used. We can see that the mobile sensor improved the tracking accuracy. In this simulation, the static sensor locations are assumed to be known, only the mobile sensor's optimal position (where the mobile sensor should be at each time step) is estimated using the proposed algorithm in Section 2. Figure 3 shows the optimal position estimation for the mobile robot at each time step. Figure 4 shows the corresponding covariance trace value at each step. In the second simulation, we focus on the simultaneous sensor nodes localization and target tracking, see Fig. 5. Figure 6 shows the mobile robot position estimation covariance and the 95% confidence bounds. From the simulations, we can see that our method can simultaneously localize sensor nodes and target at the same time. This advantage is very novel compared to the other methods such as EKF based sensor node localization.

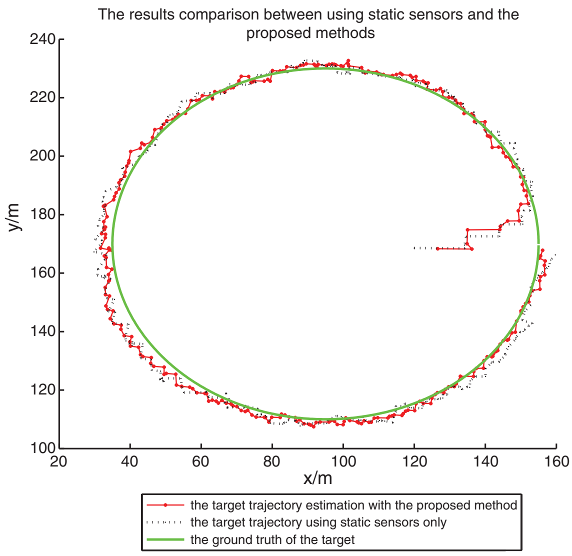

Fig. 2. Tracking results comparison with/without mobile sensor/robot.

6. Conclusions

This paper presented an on-line approach that can estimate the sensor nodes location and simultaneously localize the mobile sensor nodes together with the target. The key idea in our scheme is to control the mobile robot to an optimal position for the

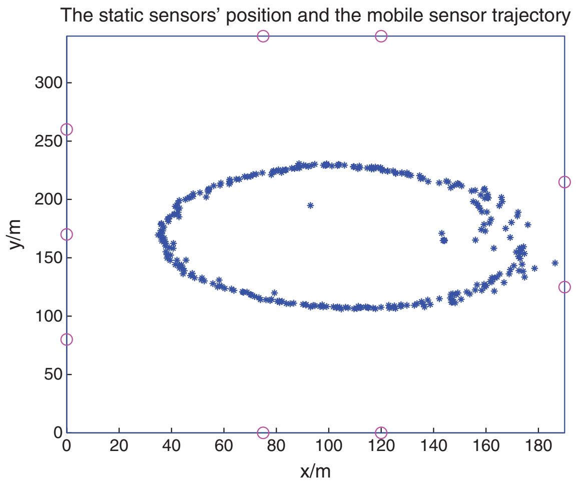

Fig. 3. The optimal position estimation for the mobile robot at each time step.

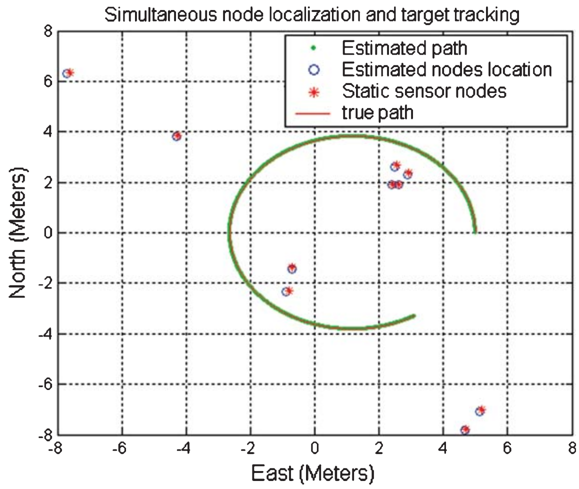

Fig. 5. Simultaneous sensor nodes localization.

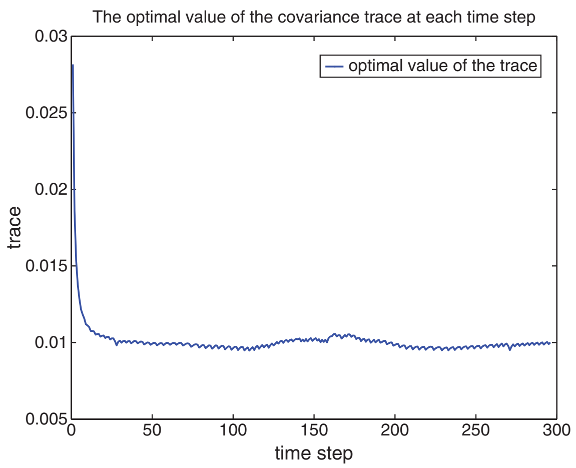

Fig. 4. The optimal value of the covariance trace.

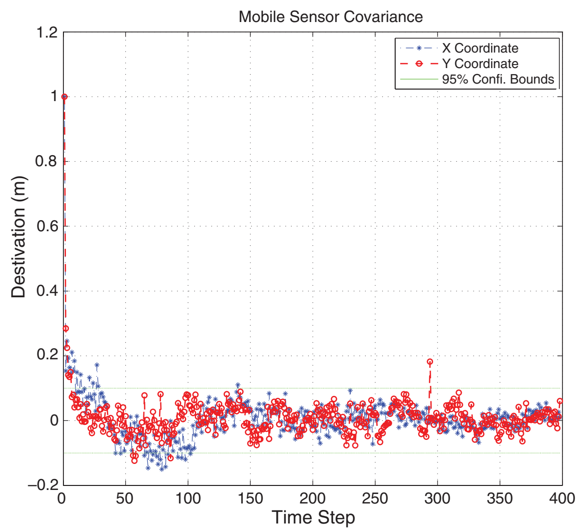

Fig. 6. Mobile robot position covariance.

best target tracking results and to use the mobile robot to simultaneously perform location estimation for the sensor nodes it passes based on the range information of the radio messages received from them. Thus, we eliminate the processing constraints of static sensor nodes and the need for static reference beacons. Our mathematical contribution is the use of an augmented extended Kalman filter (AEKF) based state estimator to solve the localization. Compared to the standard extended Kalman filter, AEKF can simultaneously localize the mobile sensor and static sensors together with the target and it is also more robust.

References

[1] Q. Hao, D.J. Brandy, B.D. Guenther, J.B. Burchett, M. Shankar and S. Feller, Human Tracking with Wireless Distributed Pyroelectric Sensors, *IIEEE Sensors Journal* **6**(6) (2006), 1683–1696.

[2] W. Xiao, L. Xie, J. Chen, L. Shue, Multi-step adaptive sensor scheduling for target tracking in wireless sensor networks, in *In the proceedings of ICASSP*, (Toulouse, France) pp. 705–708, May 2006.

[3] W. Xiao, J.K. Wu, L. Xie, Adaptive sensor scheduling for target tracking in wireless sensor network, in *Advanced Signal Processing Algorithms, Architectures, and Implementations XV, edited by Franklin T. Luk, Proc. of SPIE*, pp. 59100B1–9, July 2005.

[4] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A Survey, *Computer Networks*, **38**(4) (2002), 393–442.

[5] C.Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE **91**(8) (2003), 1247–1256.

[6] N. Patwari, A.O. Hero, III, M. Perkins, N.S. Correal, R.J. O'Dea, Relative location estimation in wireless sensor networks, *IEEE Trans. Signal Processing* **51** (2003), 2137–2148.

[7] T.C. Henderson, C. Sikorski, E. Grant, K. Luthy, Computational Sensor Networks, in Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), San Diego, USA, 2007.

[8] J. Bachrach, C. Taylor, Localization in sensor networks, in Handbook of Sensor Networks : Algorithms and Architectures, 1st ed., vol. 1, I. Stojmenovi Ed. USA: Wiley, 2005.

[9] R.L. Moses, D. Krishnamurthy, R.M. Patterson, A self-localization method for wireless sensor networks, *EURASIP Journal on Applied Signal Processing* **4** (2002), 348–358.

[10] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses, N.S. Correal, Locating the nodes: cooperative localization in wireless sensor networks, Signal Processing Magazine, IEEE **22** (2005), 54–69.

[11] K.I.M. McKinnon, Convergence of the Nelder –M ead simplex method to a non-stationary point, *SIAM J Optimization* **9** (1999), 148–158. (algorithm summary online).

[12] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, INC, 2001.