Simultaneous Adaptive Localization of a Wireless Sensor Network

Koushil Sreenath
koushils@arri.uta.edu

Frank L. Lewis
lewis@uta.edu

Dan O. Popa
popa@arri.uta.edu

Automation & Robotics Research Institute (ARRI)
University of Texas at Arlington, Arlington, Texas, USA.

A range-free approach for adaptive localization of un-localized sensor nodes employing a mobile robot with GPS is detailed. A mobile robot navigates through the sensor deployment area broadcasting its positional estimate and the uncertainty in its estimate. Distributed computationally-inexpensive, discrete-time Kalman Filters, implemented on each static sensor node, fuse information obtained over time from the robot to decrease the uncertainty in each node's location estimate. On the other hand, due to dead reckoning and other systematic errors, the robot loses positional accuracy over time. Updates from GPS and from the localized sensor nodes serve in improving the localization uncertainty of the robot. A Continuous-Discrete Extended Kalman Filter (CD EKF) running on the mobile robot fuses information from multiple distinct sources (GPS, various sensors nodes) for robot navigation. This two-part procedure achieves simultaneous localization of the sensor nodes and the mobile robot. Also presented is an adaptive localization strategy to navigate the mobile robot to the area of least localized sensor nodes. This ensures that the robot maneuvers to an area where the sensor nodes possess the largest uncertainty in location, so that it can maximize the usefulness of its positional information in best localizing the overall network.

Index Terms— Adaptive Localization, Continuous-Discrete Extended Kalman Filter (CD EKF), Simultaneous Localization, Sensor Networks.

I. Introduction

Location information is imperative for applications in both wireless sensor networks and mobile robotics. Many sensor network applications, such as tracking targets, environmental monitoring, geo-spatial packet routing, require that the sensor nodes know their locations. The large scale of deployment in sensor networks makes careful placement or uniform distribution of sensor nodes impractical. The requirement of the sensors to be small, un-tethered, low energy consuming, cheap, etc., make the sensors resource-constrained [1]. Localization is a challenging problem and yet crucial for many applications.

Approaches to the problem of localization are varied. A detailed introduction to localization in sensor networks is presented in [2]. GPS [3] solves the problem trivially, but equipping the sensors with the required hardware may be impractical. A small section of active beacons can be placed in the sensor network and other sensors can derive their location from these anchor nodes [4], [5]. Cooperative localization methods have been developed for relative localization [6], [7]. Other approaches involve RSSI [8], TOA [9], [10], AOA [11], and Signal Pattern Matching [3].

For localization with no additional hardware on the sensor node, the geometric constraints of radio connectivity are exploited. Some authors suggest using a mobile robot (whose position is known) to localize the sensors. However, the position of the mobile robot may be hard to determine. Sequential Monte Carlo localization [12] presents a range-free localization method in the presence of mobility using mobile and seed nodes. LaSLAT [13] uses a Bayesian filter to localize the sensor network and track the mobile robot. In [14], a particle filter is employed to localize elements of the network based on observation of other elements of the network. In [15], a mobile robotic sensor localized the network based on simple intersections of bounding boxes. In [16], geometric constraints based on both radio connectivity and sensing of a moving beacon

This work was supported in part by the Army Research Office grants W91NF-05-1-0314, M-47928-CI-RIP-05075-1 and the National Science Foundation grants ISS-0326595, CNS-0421282.

localize the sensor network. The Kalman filter has been used in dead-reckoning for mobile robots but its full potential in localization of WSN has not heretofore been fully explored. In [17], an extended Kalman filter is used for localization and tracking of a target. [18] uses RSSI and a robust extended Kalman filter implemented on a mobile robot to localize a delay-tolerant sensor network. The Kalman filter was used in [19] for active beacon and mobile AUV localization and in [20] for scheduling of sensors for target tracking. SLAM [21] and CML [22] employ Kalman filters for concurrent mapping and mobile robot localization, which can be considered similar to our work wherein the geometric constraints introduced due to radio connectivity of the static sensors play the role of features. In this paper we use the full capabilities of the Kalman filter in the general WSN localization problem.

Our work exploits geometric constrains based on radio connectivity such that range information is not needed. A mobile robot initially sweeps the network, and broadcasts from the robot are used to localize the sensor nodes. Computationally inexpensive Kalman filters implemented on the sensors fuse the information. On the other hand, as time passes, the mobile robot gradually loses its own localization information. We present an algorithm that uses updates from the better localized sensors along with GPS updates, when they occur, to correct this problem. A continuous-discrete extended Kalman filter running on the robot estimates the robot state continuously and fuses the discrete measurement updates.

Finally, an adaptive localization algorithm, based on adaptive sampling techniques [23], [24], is presented that navigates the mobile robot to an area of nodes with highest position uncertainty. This ensures that the robot maneuvers to an area where the nodes are least localized, so that it can maximize the usefulness of its positional information in best localizing the overall network. The adaptive localization strategy ensures that, with a minimal robot movement, the largest reduction in aggregated node uncertainty is achieved at every iteration of the adaptive localization algorithm.

The Paper is organized into the following sections. Section II presents an algorithm for localization of static sensor nodes using positional updates broadcast from the mobile robot. Section III presents an algorithm that updates the location information of the mobile robot based on GPS measurements, when they occur, and position information from nodes that are well localized. We illustrate the simultaneous localization of both static sensors and the mobile robot by fusing information from multiple sources. Section IV addresses the problem of where to send the mobile robot next to maximally decrease the localization uncertainty in the sensor network. This is the scenario of Adaptive Localization. Section V presents some discussions and future work. Section VI concludes the paper.

II. Sensor Localization using Mobile Robot

In this section we provide an algorithm that runs on each Unattended Ground Sensor (UGS) node that allows it to update its position estimate, and the uncertainty in that estimate, as a mobile robot with known position moves through the network. The algorithm is range-free in that only the communication range need be known, not the range from the node to the mobile robot. It is assumed in this section that the mobile robot's position is exactly known.

II.A. Scenario

A deployed wireless sensor network comprised of static unattended ground sensors is to be absolutely localized by a mobile robot. The robot broadcasts consist of its own position and its position uncertainty estimates. Broadcasts can only be heard within the robot's communication range. The static sensors, on receiving these broadcasts, combine the new information to update their current location estimate. A simple discrete-time Kalman filter running on each static sensor node serves to fuse information and update its location and uncertainty estimates.

This is a formalized rigorous approach employing Kalman filters for localization, in contrast to bounding boxes [15], [16], which are harder to update and keep track of. The developed algorithm is simple and can efficiently be implemented on the sensor nodes with a small computing power. The Kalman filter is simply an optimal recursive data processing algorithm [25] and has been subject of extensive research and applications, particularly in the area of autonomous navigation.

II.B. Robot Control

A classical three-wheeled tricycle robot model is employed in all simulations. This configuration uses a controlled steering angle and drive speed to navigate to a desired position as illustrated in Figure 1.

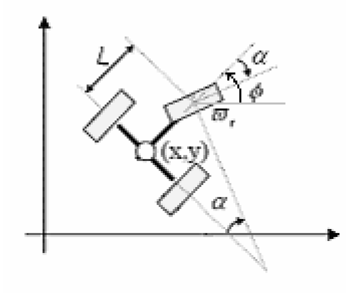


Figure 1: Tricycle Robot Configuration.

The states and kinematics of the robot are given by,

$$X = [x \quad y \quad \phi \quad \alpha]^T \tag{1}$$

$$\dot{X} = a(x,t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v_t \cos\alpha \cos\phi \\ v_t \cos\alpha \sin\phi \\ {}^{v_t}\!/_{L} \sin\alpha \\ \omega_\alpha \end{bmatrix} \tag{2}$$

with (x, y) the position of the robot, α the steering angle, and ϕ the heading angle. The control inputs are the speed v_t and the steering rate ω_α .

A simple Proportional-Derivative goal-based controller with a temporally varying goal is implemented to navigate the robot along a desired trajectory. For more details, see [26].

This dynamical setup allows more accurate simulations than the simple moving-point model usually assumed in sensor network localization papers.

II.C. Sensor Node Kalman Filter

Each static sensor node maintains its own position and uncertainty estimates. The mobile robot broadcasts contain the robot's position estimate and uncertainty estimate. The broadcasts can only be heard within the robot's communication range. A discrete-time Kalman filter running on each sensor node combines this information to optimally update the node's position estimate and its uncertainty. For more details on the derivation of the Kalman filter equations, interested readers are referred to [30].

The Kalman filter is a set of mathematical equations running in a software algorithm that provide an efficient computational means to estimate the state of a process. The state of sensor i at discrete time instant k is

$$x_k^i = \begin{bmatrix} x^i & y^i \end{bmatrix}^T \tag{3}$$

The sensor state is governed by the linear stochastic difference equation

$$x_{k+1}^i = A_k^i x_k^i + B_k^i u_k^i + G_k^i w_k^i \tag{4}$$

with measurements given by

$$z_k^i = H_k^i x_k^i + v_k^i \tag{5}$$

The random variables w_k^i and v_k^i represent process and measurement noises given by

$$x_0^i = \left(\bar{x}_0^i, P_{x_0}^i\right), w_k^i = \left(0, Q_k^i\right), v_k^i = \left(0, R_k^i\right) \tag{6}$$

where (m, P) denotes a Gaussian noise process with mean m and covariance P .

For stationary nodes, the system matrices are given by

$$A_k^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B_k^i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, G_k^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, H_k^i = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{7}$$

The *a priori* position estimates prior to measurement updates at time $k+1$ are given by the time update equations, which give the effects of time on sensor localization:

$$P_{k+1}^{i\ -} = P_k^i + Q_k^i \tag{8}$$

$$\hat{x}_{k+1}^{i\ -} = \hat{x}_k^i \tag{9}$$

In these equations, $\hat{x}_k^i$ represents the position estimate of node i at time k, while the covariance matrix P_k^i gives the corresponding uncertainty in the position estimate.

The *a posteriori* estimates given a position measurement z_k are given by the measurement update equations, which gives the effect of the robot broadcast on sensor localization:

$$P_{k+1}^i = \left[P_{k+1}^{i\ -\ -1} + H_{k+1}^{i\ T} R_{k+1}^{-1} H_{k+1}^i \right]^{-1} \tag{10}$$

$$\hat{x}_{k+1}^i = \hat{x}_{k+1}^{i\ -} + P_{k+1}^i H_{k+1}^{i\ T} R_{k+1}^{-1} \left(z_{k+1}^i - H_{k+1}^i \hat{x}_{k+1}^{i\ -} \right) \tag{11}$$

The covariance matrices Q_k^i and R_k are design parameters chosen by the engineer. With a zero Q_k^i , the uncertainty in location of the sensor i remains constant with time. With an extremely small Q_k^i , the localization uncertainty slowly drifts with time. This means that the current measurements from the mobile robot are given more weight than the current node position estimate, which avoids the node's becoming too certain of a position that may be incorrect.

When the robot is in range and the sensor hears the broadcast position of the robot, the measurement update equation is used to combine the new information to improve sensor node position and uncertainty estimates. In this section, the robot is assumed to be perfectly localized. Thus when a sensor hears a broadcast, it could only be within the communication range of the robot whose position is broadcast. The measurement uncertainty matrix R_k reflects this, and is chosen as

$$R_k = \sigma^{Bot}, \sigma^{Bot} = \begin{bmatrix} \sigma_x^{Bot} & 0 \\ 0 & \sigma_y^{Bot} \end{bmatrix} \tag{12}$$

$$\sigma_x^{Bot} = \frac{Range_x^{Bot}}{\sigma_{const}}, \sigma_y^{Bot} = \frac{Range_y^{Bot}}{\sigma_{const}} \tag{13}$$

where σ^{Bot} is the uncertainty introduced due to $Range^{Bot}$, the communication range of the robot. We assume the design parameter $\sigma_{const} = 3$, to include 70% of the communication range, $Range^{Bot}$, of the robot. (Gaussian uncertainties are assumed.) Through this selection of R_k the Kalman filter automatically takes care of the range of the robot within which it hears broadcasts.

Algorithm 1 shows the position update algorithm that runs on each node, which is very simple and easy to implement. It consists of four equations, two for the time update, and two for the measurement update. This algorithm automatically provides uncertainty estimates through the computation of the error covariance P_k^i, which is equivalent to the bounding box information provided by the algorithm in [15].

Algorithm 1: Static sensor node localization algorithm

```
1.  At each discrete time instant,
2.  if  robot broadcast received by sensor
3.  then
4.    Update   sensor   state   and   uncertainty
    estimates using KF measurement Eqs. (10,11).
5.  else
6.    Propagate estimates using time update Eqs.
    (8,9).
7.  end if
```

II.D. Simulation Results

Extensive simulations have been performed to verify the effectiveness of the proposed algorithm. We also studied the effects of initial sweep paths and the robot broadcast interval on sensor localization. The mobile robot is navigated along the desired sweep path and periodic location information is broadcast. On receiving a broadcast, sensors update their location and uncertainty estimates. This is a range-free procedure that relies on the limited communication range of the robot, and as such, the sensor locations are updated based on the position of the robot. That is, the updated sensor position estimate is a weighted combination of its current location estimate and the current location of the robot. Thus sensors hearing only one broadcast will have an estimated location that is projected onto the path of the robot.

Figure 2 shows the initial sinusoidal sweep path and the position and range of the broadcast with a broadcast interval of 5 discrete time instants. The desired path is the setpoint trajectory input to the robot controller, and the simulated path is the path along which the controller drives the robot. These paths differ due to the non-holonomic kinematics of a tricycle configuration. Robot broadcasts consist of locations from the simulated path in this scenario. The 'x' represent the actual positions of the static sensors that are to be localized. The sensor nodes are randomly deployed in an area of 100x100 units and do not initially know their actual positions. The nodes all have initial position estimates being the centroid of the deployment area, and an initial uncertainty of infinity, corresponding to complete lack of knowledge of their positions. The sensors have a communication range of 20 units (corresponding to a 20m range of a MICA mote), and the robot has a communication range of 15 units.

Figure 3 illustrates the localized sensors after the robot has made its sweep through the network. The '•' represent the final position estimates of the nodes.The uncertainty of the sensors in their position estimates has been depicted as circles and ellipses representing 3σ of the uncertainty distribution, assuming Gaussian uncertainties. Note that the sensors always outside the communication range of the mobile robot do not become localized. The sensors that receive more than one broadcast from the mobile robot end up being better localized, since each position update reduces the position uncertainty.

The effectiveness of the algorithm is demonstrated by the fact that in every case, the actual location (marked by an 'x') is within the uncertainty bound of the estimated position (marked by a '•').

The localization error of the sensors, computed as the Euclidean distance between true and estimated positions, is depicted in the vertical axis of Figure 4. Sensors near the path of the mobile robot that have received multiple broadcasts have smaller errors.

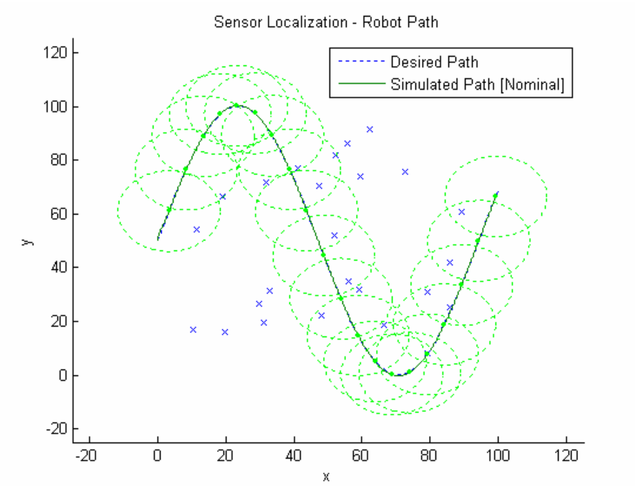


Figure 2: Initial sinusoidal sweep path with broadcast locations and range of broadcast.

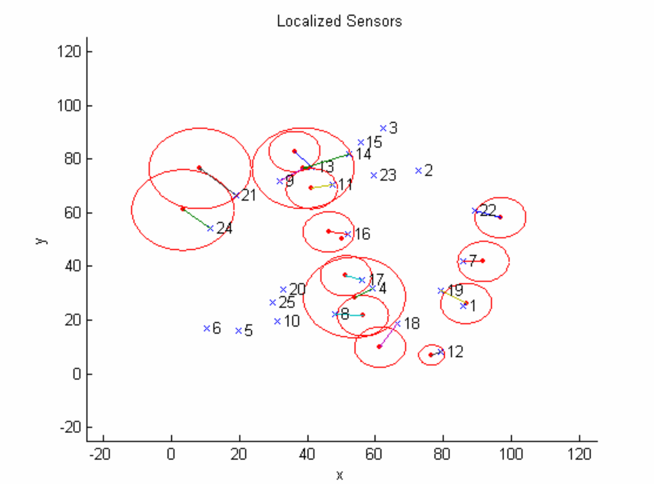


Figure 3: Localized sensors, real positions (denoted by ‘x’) and estimated positions (denoted by ‘●’), are illustrated after initial mobile robot sweep of the deployment area. Uncertainty rectangles have been illustrated to depict the uncertainty of the sensor in its position estimate.

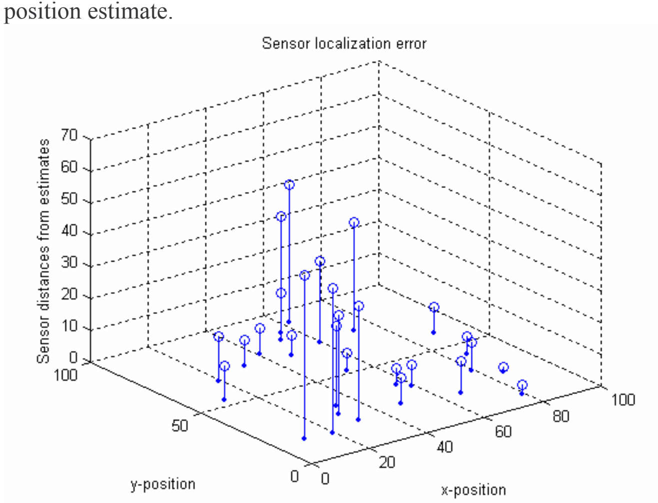


Figure 4: Localization error, computed as the Euclidean distance between real and estimated positions, of sensors after initial sweep of the deployment area.

The same simulation was rerun with different mobile robot broadcast intervals, and the effect of broadcast interval on the average localization error of the network is depicted in Figure 5. Generally, as broadcast interval decreases, the average error decreases.

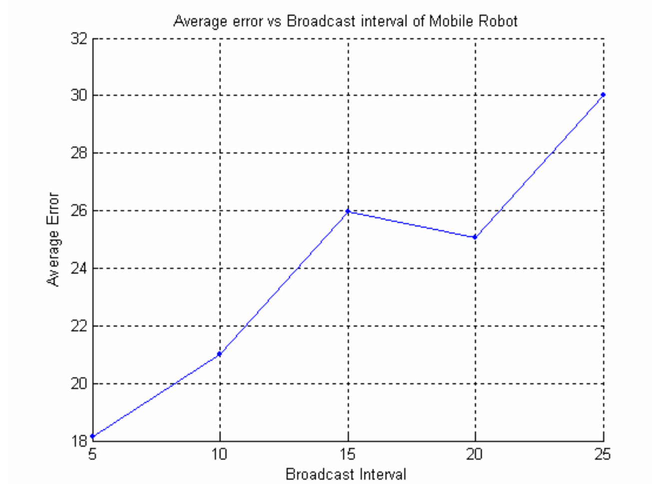


Figure 5: Effect of broadcast interval on average localization error.

III. Simultaneous Mobile Robot and Sensor Localization

In this section we consider the realistic case where the mobile robot’s position is not exactly known. We provide an algorithm which runs on the mobile robot that fuses position information from GPS, when it is available, and from the already-localized sensor nodes. This allows the robot to update its position estimate as well as the uncertainty estimate. When this algorithm is run simultaneously with the algorithm of the previous section running on each sensor node, the result is simultaneous mobile robot and sensor localization. A procedure is given to avoid detrimental recursive feedback between the two algorithms.

III.A. Mobile Robot Localization

When localizing the sensor nodes in the previous section, the robot was assumed to know its position exactly at all instants of time. However, as the robot navigates by dead reckoning, or due to steering inaccuracies, its localization increasingly deteriorates as time passes. Location updates from the GPS, when they occur, and from stationary sensor nodes that have already been localized can be used to improve the localization estimate of the robot.

Some sensor nodes are localized more finely due to more numerous updates they have previously received from the mobile robot. These sensors can be employed to localize the robot when its position information deteriorates. This is accomplished by having each sensor node make a transmission that contains the node’s position estimate and uncertainty. This is received by the robot when it is in range. The sensors transmit at fixed intervals, with each sensor having a different random interval. This ensures that the updates between mobile robot and sensor nodes are staggered in time and that no recursive feedback occurs. A recursive feedback practically feeds back to the robot, it's own positional estimate and a slightly larger uncertainty estimate than its own. This recursive measurement update does not change the positional estimate, but however wrongly reduces the robot's uncertainty estimate. A similar case arises for the sensor if it immediately receives yet another update from the robot.

A continuous-discrete extended Kalman filter running on the mobile robot is used to simulate the robot and update the states using measurements from the GPS system and the better-localized UGSs. Extended Kalman filters have been used for local and infrequent global senor data fusion [27], for mobile robot localization [28], and in navigation of autonomous vehicles [29]. For information about the Extended Kalman filter see [30].

The continuous-time system model of the robot is given by (2) as

$$\dot{X} = a(X,u,t) + G(t)w \tag{14}$$

The sampled discrete-time measurement model of the robot is given by

$$\begin{aligned} Z_k^{gps} &= h^{gps}\left[X(t_k),k\right] + v_k^{gps} \\ Z_k^{ugs} &= h^{ugs}\left[X(t_k),k\right] + v_k^{ugs} \end{aligned} \tag{15}$$

where

$$X(0) = \left(\overline{X}_0, P_0\right), w(t) = (0,Q), v_k^{gps} = \left(0, R^{gps}\right), v_k^{ugs} = \left(0, R^{ugs}\right) \tag{16}$$

$$a(X,t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v_t \cos\alpha\cos\phi \\ v_t \cos\alpha\sin\phi \\ v_t/L \sin\alpha \\ \omega_\alpha \end{bmatrix}, G(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

$$h^{gps}\left[X(t_k),k\right] = \begin{bmatrix} x \\ y \end{bmatrix}, h^{ugs}\left[X(t_k),k\right] = \begin{bmatrix} x \\ y \end{bmatrix} \tag{18}$$

In the extended Kalman filter, the effect of time on the robot states is given by the time update equation

$$\begin{aligned} \dot{\hat{X}} &= a\left(\hat{X},u,t\right) \\ \dot{P} &= A\left(\hat{X},t\right)P + PA^T\left(\hat{X},t\right) + GQG^T \end{aligned} \tag{19}$$

In [31], the deleterious effects of time passing are shown in terms of increasing position uncertainty and decreasing belief. These effects are formally captured in a rigorous manner by the time-update equations (18)-(19), which shows how uncertainty increases due to dead reckoning and steering uncertainties.

The effects of the GPS navigation updates, when they are received, are given by the measurement update equation

$$\begin{aligned} K_k &= P^-(t_k)H^{gps^T}\left(\hat{X}_k^-\right)\left[H^{gps}(\hat{X}_k^-)P^-(t_k)H^{gps^T}(\hat{X}_k^-) + R^{gps}\right]^{-1} \\ P(t_k) &= \left[I - K_k H^{gps}\left(\hat{X}_k^-\right)\right]P^-(t_k) \\ \hat{X}_k &= \hat{X}_k^- + K_k\left[Z_k^{gps} - h^{gps}\left(\hat{X}_k^-,k\right)\right] \end{aligned} \tag{20}$$

The effects of the updates based on localized sensor nodes, when they are received, are given by the UGS measurement update equation

$$\begin{aligned} K_k &= P^-(t_k)H^{ugs^T}\left(\hat{X}_k^-\right)\left[H^{ugs}(\hat{X}_k^-)P^-(t_k)H^{ugs^T}(\hat{X}_k^-) + R^{ugs}\right]^{-1} \\ P(t_k) &= \left[I - K_k H^{ugs}\left(\hat{X}_k^-\right)\right]P^-(t_k) \\ \hat{X}_k &= \hat{X}_k^- + K_k\left[Z_k^{ugs} - h^{ugs}\left(\hat{X}_k^-,k\right)\right] \end{aligned} \tag{21}$$

The measurement uncertainty matrices R^{gps} and R^{ugs} represent the uncertainty in the GPS and the uncertainty in the update offered by UGS i respectively. The uncertainty in the sensor update, R^{ugs}, is a combination of the uncertainty of the sensor position and the uncertainty due to the communication range of the sensor. These uncertainties combine in quadrature as

$$\begin{aligned} R_k^{ugs} &= \left[P^{i\,2} + \sigma^{i\,2}\right]^{1/2}, \sigma^i = \begin{bmatrix} \sigma_x^i & 0 \\ 0 & \sigma_y^i \end{bmatrix} \\ \sigma_x^i &= \frac{Range_x^i}{\sigma_{const}}, \sigma_y^i = \frac{Range_y^i}{\sigma_{const}} \end{aligned} \tag{22}$$

where σ^i is the uncertainty introduced due to $Range^i$, the communication range of sensor i.

Similarly, the measurement noise covariance of the sensor, eq. (12), has to be modified to include the uncertainty in the robot's position. The robot is no longer absolutely localized with zero uncertainty. The uncertainty in robot localization and the uncertainty due to robot communication range combine in quadrature, modifying eq. (12) as

$$R_k = \left[P_{XY}^{Bot\,2} + \sigma^{Bot\,2}\right]^{1/2} \tag{23}$$

P_{XY}^{Bot} is the partial error covariance of the robot which effects only the position of the robot, and σ^{Bot} is as defined earlier.

The Jacobians of the nonlinear system, determined from (2), are given by the following system matrices:

$$\begin{aligned} A(X,t) &= \frac{\partial a(X,t)}{\partial X} = \begin{bmatrix} 0 & 0 & -v_t\cos\alpha\sin\phi & -v_t\sin\alpha\cos\phi \\ 0 & 0 & v_t\cos\alpha\cos\phi & -v_t\sin\alpha\sin\phi \\ 0 & 0 & 0 & v_t/L\cos\alpha \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ H^{gps}(X) &= \frac{\partial h^{gps}(X,k)}{\partial X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ H^{ugs}(X) &= \frac{\partial h^{ugs}(X,k)}{\partial X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned} \tag{24}$$

With these equations in place and programmed as a software algorithm on the mobile robot, and the sensor nodes running the algorithm presented in the previous section, the mobile robot and the static sensors automatically mutually update their estimates with incoming updates. There is no additional decision-making logic to be implemented as in other range-free work discussed earlier. There is no need to compute bounding boxes, as the error covariance matrices are automatically updated as measurements are received.

The algorithm to be implemented on the mobile robot that updates its position estimate and uncertainty based on GPS measurements and on the localized sensor nodes is given as Algorithm 2. This algorithm is efficient to implement since the bulk of it is mathematical equations.

When Algorithm 2 is run on the robot simultaneously along with Algorithm 1 on each sensor node, simultaneous mobile robot and sensor localization occurs.

Algorithm 2: Mobile robot localization algorithm.

```
1.  Navigate robot along desired path.
2.  Broadcast location information at discrete
    intervals.
3.  if broadcast from GPS received
4.    Update robot state and uncertainty
    estimates using measurement Eq. (20).
5.  end if
6.  if broadcast from sensor received
7.    Update robot state and uncertainty
    estimates using measurement Eq. (21).
8.  end if
```

III.B. Simulation Results

The simulations described in Section II have been rerun with GPS updates and sensor updates implemented as Algorithm 2 on the mobile robot. Infrequent GPS updates and temporally staggered sensor updates help localize the robot. Figure 6 shows the robot's sweep path with GPS and UGS updates disabled. A systematic dead reckoning error [32] has been injected into the mobile robot to give gradually deteriorating position information. The localization of the robot deteriorates with time as can be seen in the deviation of the robot's estimated path (hyphenated green line) from the robot's true path (continuous red line.) The desired path is as defined earlier, the simulated path is the estimated path of the robot (where the robot thinks it is), and the actual path is the true path of the robot. Robot broadcasts occur along the true path but the broadcasted position is a corresponding location on the estimated path of the robot. If not corrected, these wrong broadcasts could incorrectly localize the sensors.

Figure 7 illustrates the robot's sweep path which is corrected in time by GPS and UGS updates using Algorithm 2. As is evident, the robot's localization has improved and the positions of where the robot thinks it is (the estimated position), and where the robot actually is (the true position) are much closer, since the estimates are continuously corrected using Algorithm 2 as position information arrives, either from GPS or from sensor node broadcasts.

Robot broadcasts occur along the true path of the robot and consist of the robot's estimated position (slightly different from the robot's true position where the broadcast occurs) and uncertainty. Sensors within range receive the broadcast and update their positional information based on the robot's estimates.

Figure 8 illustrates the localized sensors after the initial sweep. True sensor positions are indicated by an 'x' and estimated positions by a '•'. Now, some true sensor positions are outside the 3σ boxes due to the added uncertainty in the robot position, though they are generally close to these boxes. Figure 9 depicts the final localization error of each sensor.

Figure 10 compares the average localization error of the sensor network obtained by running algorithm 1 in section I and algorithm 2 in section II with UGS and GPS updates off and on. As can be seen, for the case of algorithm 1 where the robot is assumed to be perfectly localized, the average localization error is less than that for algorithm 2. And in the case of algorithm 2, better localization is obtained when the robot receives updates from the UGSs and the GPS system.

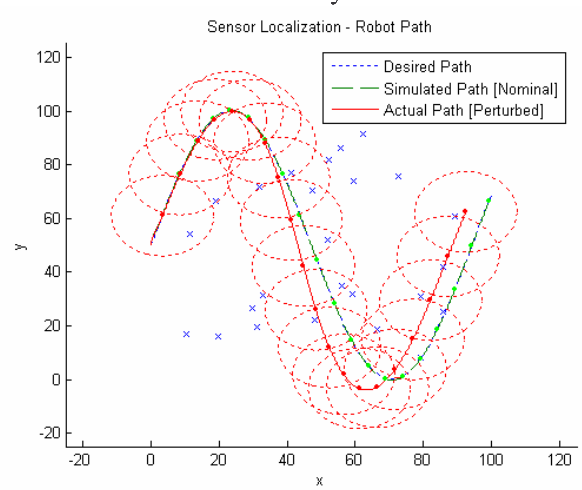


Figure 6: Initial sweep path of the robot with GPS and UGS updates disabled. Robot's localization deteriorates with time as evident in the deviation in the estimated path (hyphenated green line) and the true path (continuous red line.)

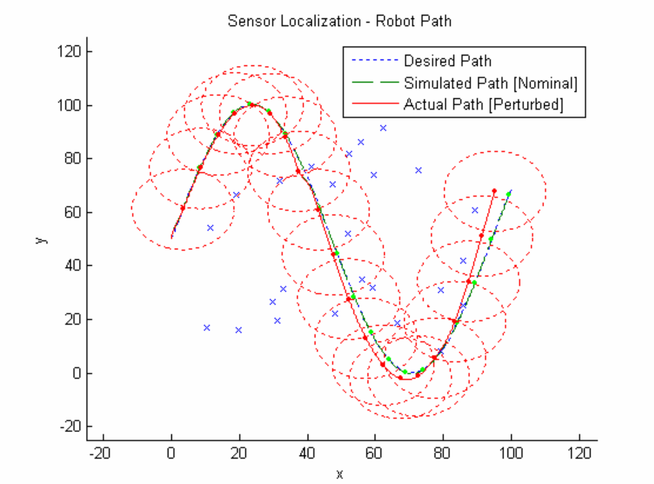


Figure 7: Initial sweep path of the mobile with GPS and UGS updates enabled as Algorithm 2. The robot's localization has improved and the true position and the estimated position of the robot along the path are much closer.

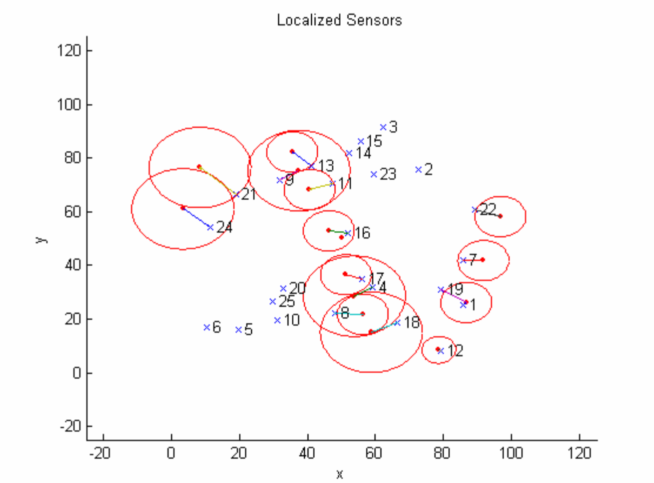


Figure 8: Localized sensors after initial sweep of the deployment area. True sensor positions are indicated by a 'x' and estimated positions by a '●'.

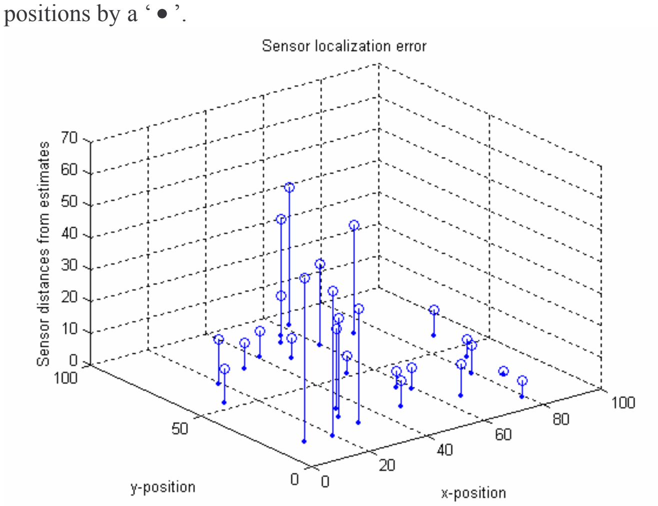


Figure 9: Localization error of sensors computed as the Euclidean distance between true and estimated positions.

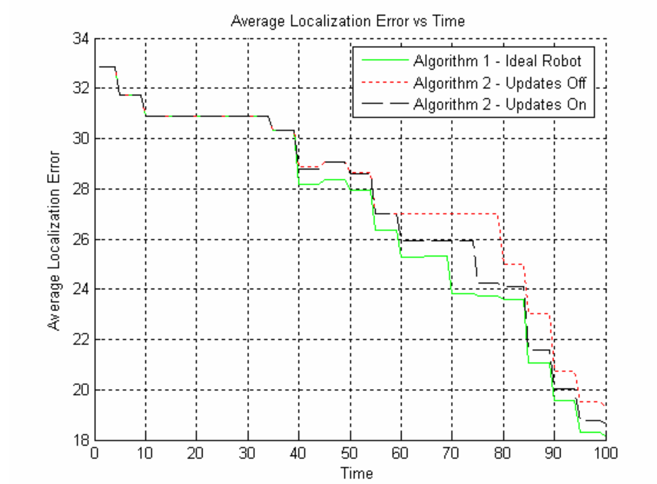


Figure 10: Average localization error over time for Algorithm 1 and for Algorithm 2 with UGS and GPS updates Off and On.

IV. Adaptive Localization

A navigation strategy, to be used subsequent to the initial sweep of the deployment area that was presented in the previous sections, is developed here which further minimizes the localization uncertainty of the sensor network in the most efficient manner. An adaptive localization policy is adopted to navigate the mobile robot to an area of least localized sensor nodes. This ensures that the robot maneuvers to an area with sensor nodes possessing the largest uncertainty in location.

Accurate position of coarsely localized sensors can not be known (due to inherent coarse localization) so that navigating to these sensors is not possible. The radio connectivity of the network is exploited to address the problem of having the robot navigate to a location which is imprecise. Figure 11 depicts the communication connectivity of the network.

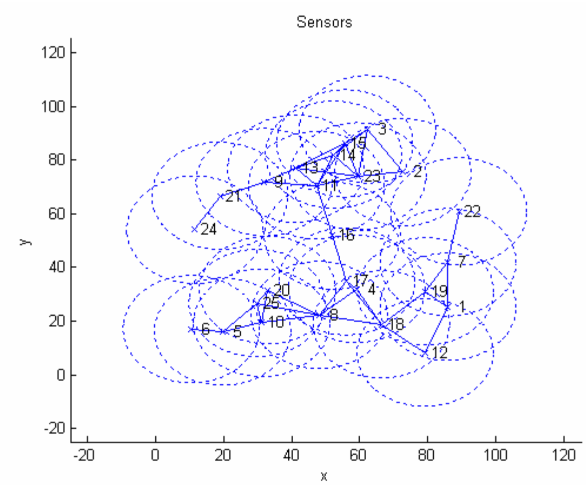


Figure 11: Communication connectivity of the network. Communication routes between sensors and range of communication of each sensor are depicted.

A communication protocol is developed wherein, the robot broadcasts a navigation request packet, *NAV-*

REQ, when the robot wants to find a new location to navigate to. Sensors which receive the *NAV-REQ* packet, forward it along the network. Sensors having a large uncertainty scalar, the Frobenius Norm [33] of the uncertainty matrix, reply back with a localization request packet, *LOC-REQ*. The *LOC-REQ* packet consists of the uncertainty matrix of the requesting sensor and propagates along the network until it is received by a friendly localized neighbor. Friendly localized neighboring sensors receiving the *LOC-REQ* packet append it with their position and forward the packet along the sensor network to the robot. Figure 12 and Figure 13 show the flow of the *NAV-REQ* and *LOC-REQ* packets.

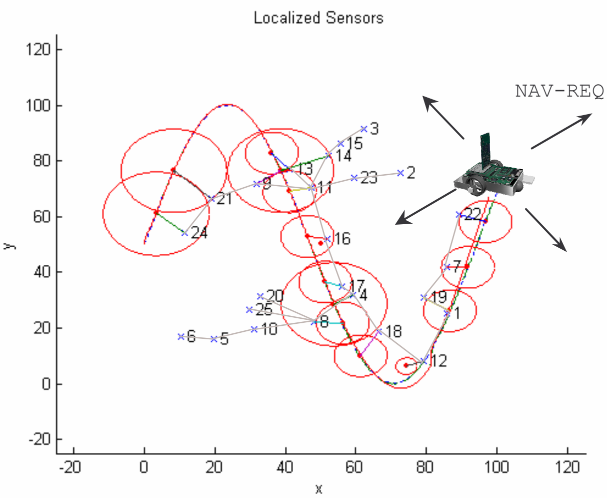


Figure 12: Initiation of the navigation request "*NAV-REQ*" packet from the robot.

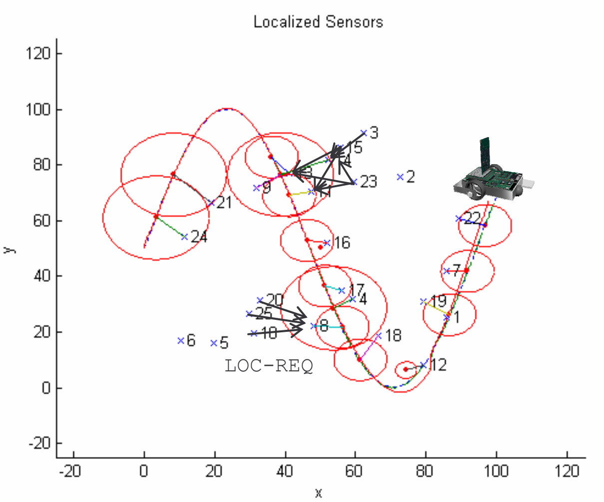


Figure 13: Coarsely localized sensors reply back with a localization request "*LOC-REQ*" packet. Black arrows indicate the flow of the "*LOC-REQ*" packets to friendly neighbor nodes which append the packet with their position and forward it along the network towards the robot.

The robot receives packets from multiple non-unique friendly neighbors each representing a single coarsely localized sensor. The robot needs to choose a friendly neighbor to navigate to. Friendly neighbor arbitration is performed by grouping uncertainties of the same friendly neighbor in quadrature to give its combined uncertainty scalar. The friendly neighbor with the largest combined uncertainty scalar is picked as the location to navigate to. If multiple such neighbors exist, the most localized neighbor is chosen. The tradeoff between combined uncertainty versus localization certainty of a friendly neighbor even when there is no tie is such that the robot always navigates to reduce the most amount of uncertainty with each iteration. This may result in more coarsely localized sensors than one or two finely localized sensors.

Thus regions with a large density of coarsely localized sensors having a common friendly neighbor are adaptively navigated to. However, due to the inherent imprecise location of the friendly neighbor, the robot actually navigates a circular path around the neighbor's estimated position.

Algorithm 3 summarizes the Adaptive localization algorithm.

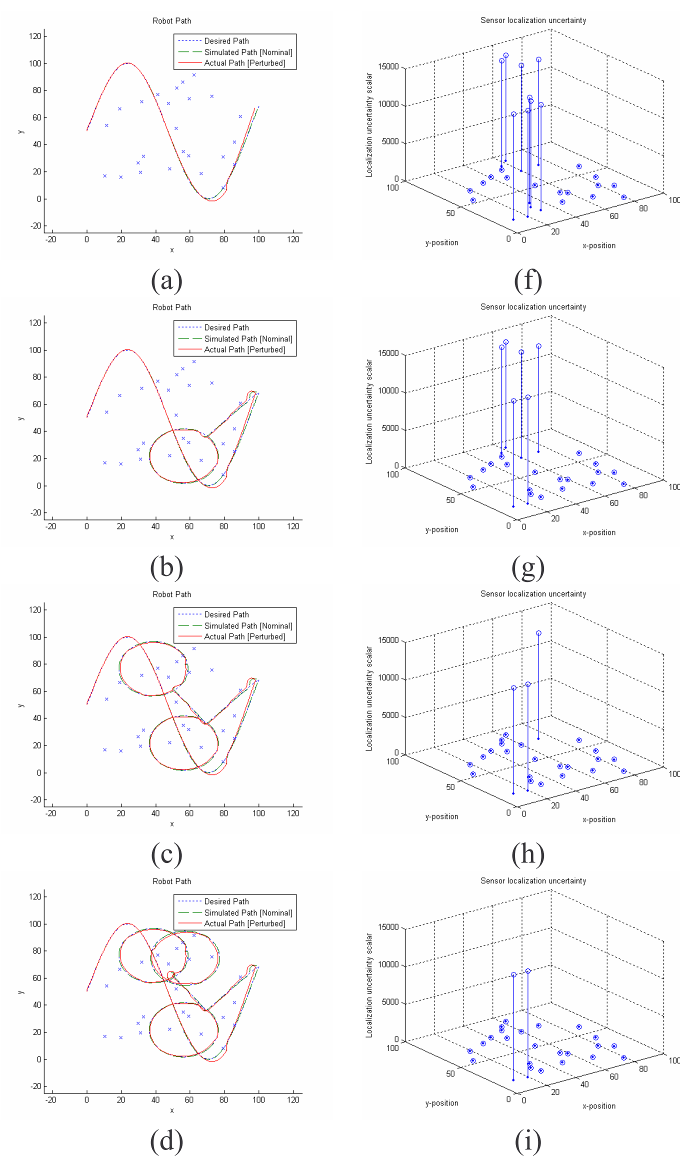
Algorithm 3: Adaptive localization algorithm.

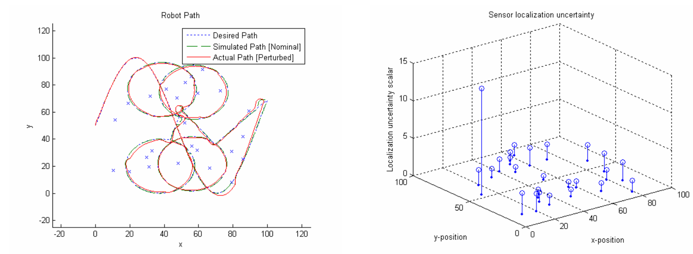
```
1.  Broadcast Navigation request, NAV-REQ,
    packet.
2.  Wait to receive Localization request, LOC-
    REQ, packets.
3.  for all LOC-REQ with the same friendly
    neighbor
4.    Combine uncertainty scalars of the
    requesting sensors.
5.  end for
6.  Pick friendly neighbor with maximum combined
    uncertainty scalar of the requesting sensors.
7.  if multiple maximas arise
8.    Among the maxima, pick the most localized
    friendly neighbor.
9.  end if
10. Navigate around the picked friendly neighbor
    executing the simultaneous localization
    algorithm, Algorithm 1 on the sensors and
    Algorithm 2 on the mobile robot.
11. Repeat Steps 1-10 as required.
```

After the initial sinusoidal sweep, see Figure 8, Figure 13, sensors 8 and 14 both receive three Localization request packets each from sensors 10,20,25, and 3,15,23 respectively. And on combining the uncertainties of the requesting coarsely localized sensors, an equal maximum uncertainty scalar arises for sensors 8 and 14. However sensor 8 is more localized than sensor 14 and robot navigation occurs around the estimated position of sensor 8, see Figure 14(b).

Figure 14 illustrates four adaptive localization iterations and its navigation path is shown in (a-e) and corresponding uncertainty scalars of the sensors at the end each adaptive localization iteration is illustrated in (f-j). With each adaptive localization iteration, Figure 15 shows the reduction of localization error of each sensor, and Figure 16 depicts the reduction of the average localization error of the sensor network. Figure 17 illustrates the localized sensors after four iterations of the adaptive localization algorithm. As can be seen, all sensors are localized and uncertainty in localization fairly small. Figure 18 compares the average localization error for the three algorithms (1-3) over time. As can be seen, with the adaptive navigation (algorithm 3), there is a significant reduction in localization error.

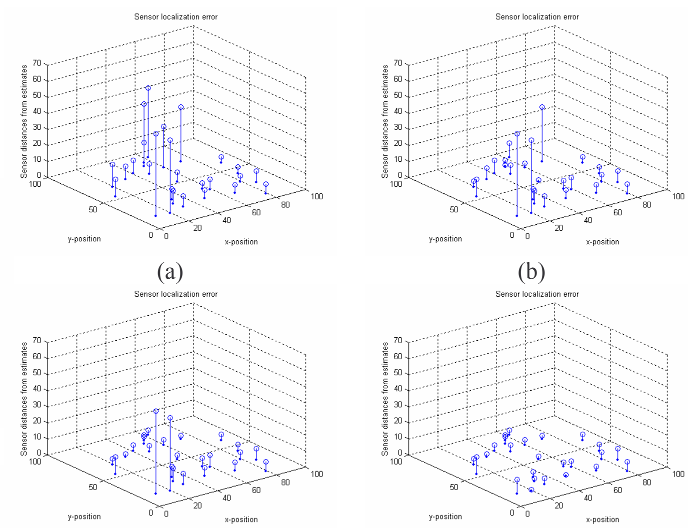
At every instant, along with the adaptive localization algorithm, Algorithm 3, the entire simultaneous localization algorithm with updates from the GPS, and more localized sensor, Algorithm 1 and Algorithm 2, are always running. This demonstrates simultaneous adaptive localization of the sensor network.





(e) (j)

Figure 14: Various adaptive localization steps (a-e) and corresponding uncertainty scalars for the sensors (f-j) after each navigation step.



(c) (d)

Figure 15: Reduction of the average Localization error of the sensors with each adaptive localization iteration.

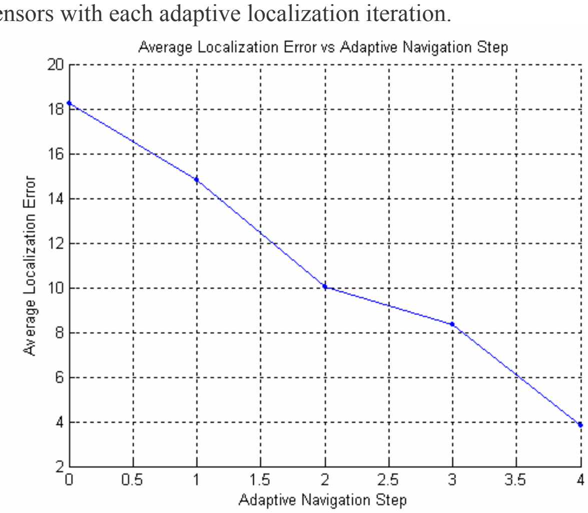


Figure 16: Reduction of average localization error with each adaptive localization iteration.

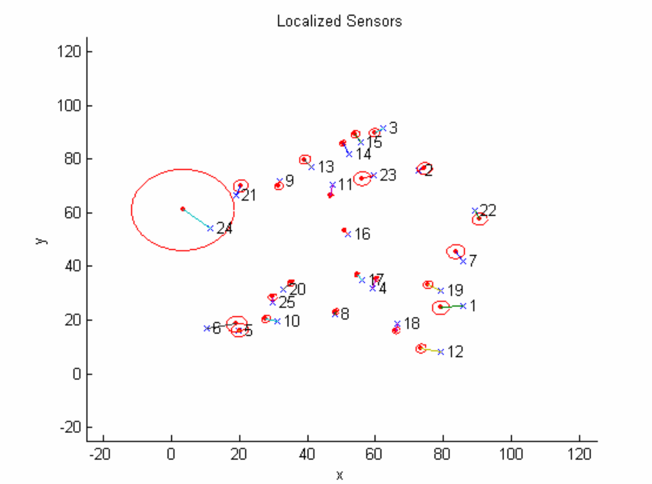


Figure 17: The final position estimates of the Localized sensors after four iterations of adaptive localization algorithm.

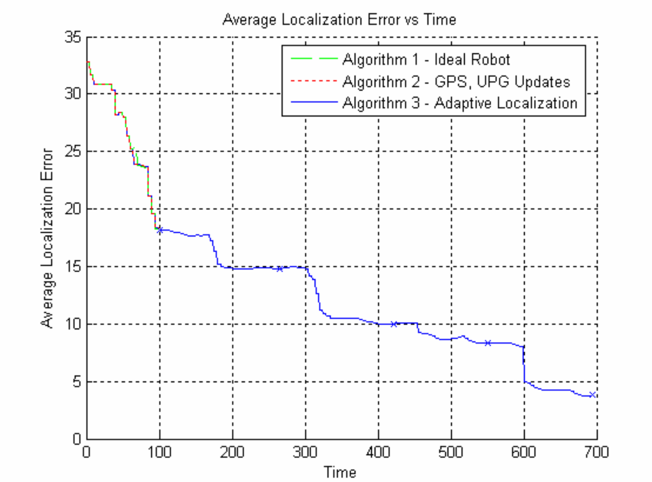


Figure 18: Average localization error of the sensor network with time for Algorithms 1, 2, and 3. A '*' demarcates each adaptive localization iteration region.

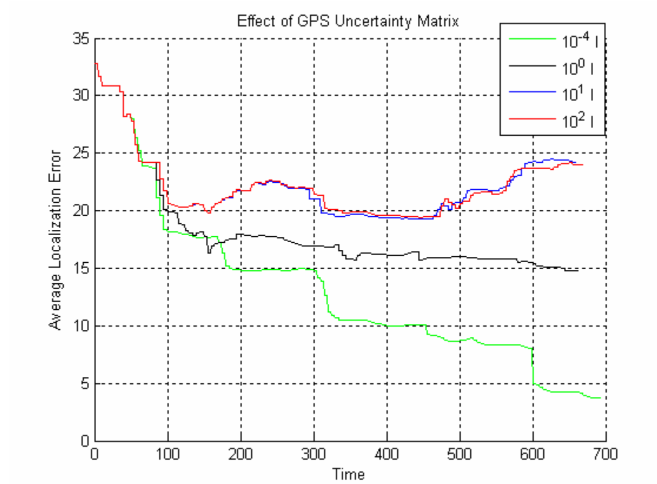
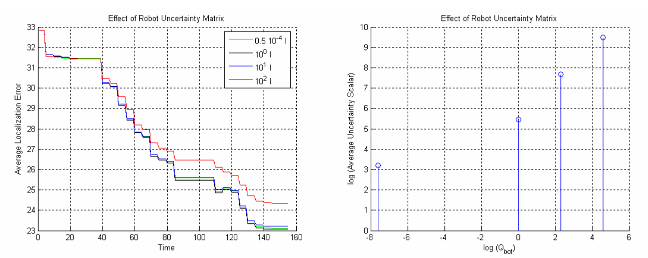


Figure 19: Effect of GPS uncertainty matrix on sensor network localization.



(a) (b)

Figure 20: Effect of Robot uncertainty matrix on (a) Average sensor network localization, and (b) Mean uncertainty scalar in sensor network localization.

V. Discussions

This section discusses how the results of this work are affected by dropping certain ideal assumptions.

V.A. Effect of uncertainty matrices

Infrequent accurate GPS updates are assumed to exist which, along with updates from the UGSs help in localizing the robot. Figure 19 shows the effect of the GPS uncertainty matrix, R^{GPS} (Eq. 20). With an highly inaccurate GPS (large R^{GPS}), the localization of the sensor network initially decreases as sensors get localized, but then increases again as the robot drifts are wrongly corrected and the sensors get localized to the increasingly incorrect robot broadcasts.

On the other hand, with increasing robot uncertainty matrix, the robot broadcasts locations are accurate but with large uncertainty. Figure 20 shows the effect of the robot uncertainty matrix through Q (Eq. 19) on localization. As expected localization error is very slightly affected, however the uncertainty of the sensors in their estimates increases with Q.

V.B. Effect of radio range and irregularity

The localization algorithms presented here entirely rely on the sensor being able to hear robot broadcasts. Figure 21 shows the effect on localization by varying the communication range of the robot only (the range of the sensors is fixed at 20 units.) An extremely small robot range does not actually improve localization as sensors either hear a small number of robot broadcasts or none at all. With large robot ranges, all sensors are rapidly localized but with coarse localization. A robot communication range that is slightly smaller than the sensor communication range seems to obtain the best localization results.

Radio irregularity is a common and non-negligible phenomenon in wireless sensor networks. It results in irregularity in radio range in different directions [34]. A Radio Irregularity Model (RIM) discussed in [35] has been implemented to see the effect of radio irregularity on sensor network localization. Figure 22(a) shows the robot communication range for different values of degree of irregularity (DOI). Figure 22(b) shows the robot broadcasts with the implemented dynamic RIM where the range is asymmetric and it dynamically reorients based on the orientation of the robot.

Algorithms 1-3 have been run ten times for each of the different DOI and the mean sensor network localization error is obtained. The number of adaptive localization iterations has been limited to four to provide means of a fair comparison. As seen in Figure 23, with increasing radio irregularity, the localization error deteriorates. Certain robot broadcasts may be lost due to the radio irregularity; however adaptive localization continues to work as long as the network does not break up into trivial clusters of a single node. For an increased DOI, the algorithm will just take more iterations to converge to an equivalent localization error obtained with no radio irregularity.

One aspect that has not so far been addressed is related to the issue of drop packets / link connectivity. In other words what happens if NAV-REQ and LOC-REQ packets are received intermittently. Research [40] has shown that nodes may be in communication range but yet occasionally drop packets due to environmental factors, interference, congestion, and other sources of loss. It's possible to keep retransmitting packets and increase the delivery probability, but this also increases the transmission time and energy consumed. Figure 24 shows the effect on the adaptive localization algorithms when communication links fail with certain probabilites and packets are dropped. The mean sensor localization error deteriorates as links fail and packets get dropped.

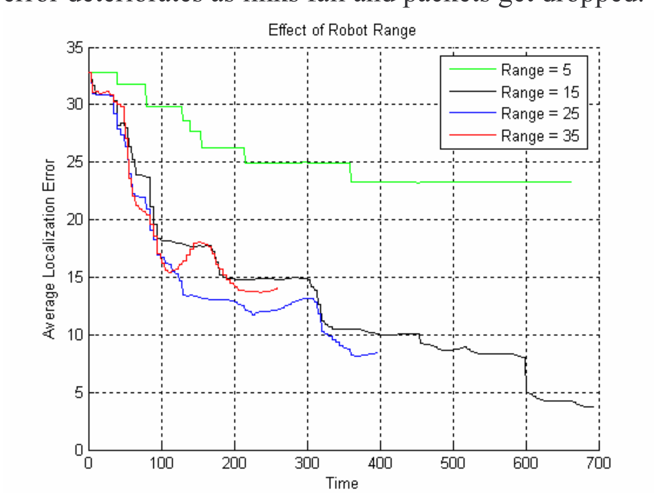
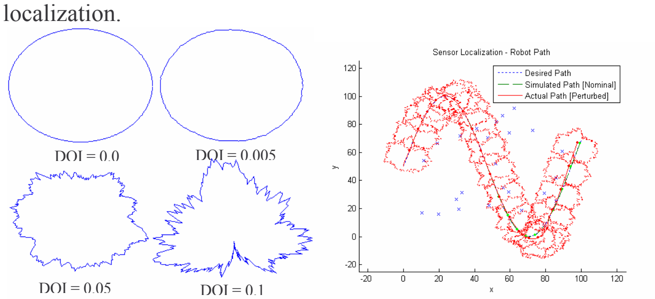


Figure 21: Effect of robot communication range on sensor network localization.



(a) (b)

Figure 22: (a) Radio communication range for different DOI, (b) Robot broadcasts with the dynamic RIM.

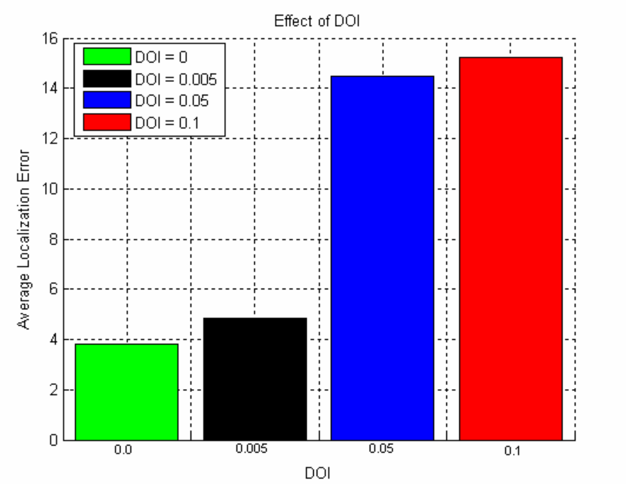


Figure 23: Effect of radio irregularity on sensor network localization.

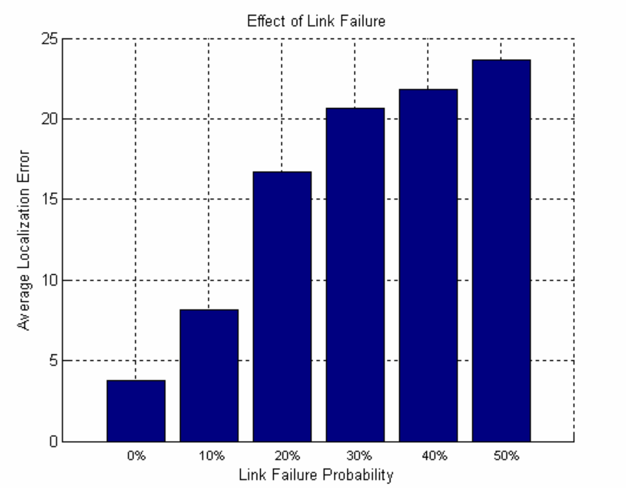


Figure 24: Effect of link failure on sensor network localization.

V.C. Energy considerations

Algorithm 1 on the sensor node, although computationally inexpensive, places the sensor in listen mode for extended periods of time. This is a case of idle listening, were the sensor node actively listens for potential robot broadcasts. Idle listening of the radio has been identified as a major source of energy wastage [36]. A survey of energy saving mechanisms in sensor networks exists in [37].

In our application, since all communication for our algorithms are discrete in nature, the listening of the nodes can be discretized. For instance, if the knowledge that the robot broadcasts every 5 seconds is available to every node, then the node can sleep for 4 seconds and listen for 1 second. This in itself is an energy saving of over 80%. All NAV-REQ, and LOC-REQ packet transmissions can by synced to this interval. Periodic correction of clock drifts for various nodes would however be required. In the initial case of a node not knowing when to listen, it can listen 100% of the time until it finds out the discrete broadcast periodic interval. For detailed information on sleep scheduling refer [38],

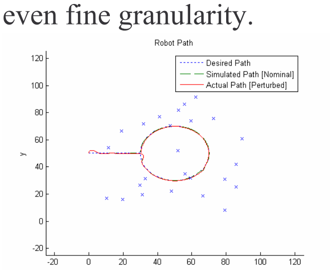
[39].

V.D. Extensions to Simultaneous adaptive localization

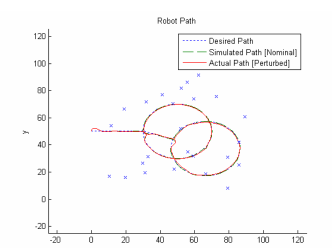
Sensor nodes are localized only by broadcasts from the mobile robot. Other sensor broadcasts could be used for localization, but for even a reasonably fine localization, multiple localized neighbors around the sensor are required. Multiple broadcasts from one particular sensor provide as much information as just one broadcast.

The initial sweep path of the mobile robot over the sensor network may place certain restrictions (such as knowing the extent of the sensor network to be swept). However, this sweep path need not cover the entire area, infact the sweep path can be entirely dropped and the adaptive localization algorithm made to completely take over. An initial NAV-REQ would result with a LOC-REQ and will navigate the robot to the initial estimate of the sensor requesting localization (usually the center of the deployment area, but could be any other location.) Figure 25 shows four adaptive localization iterations with no initial network sweep. As shown in Figure 26, with no initial sweep, the localization error initially is high, but eventually converges to that obtained with an initial sine sweep.

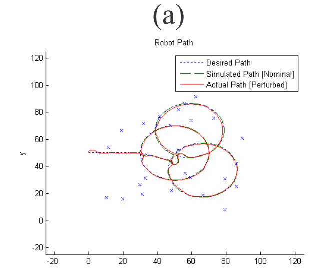
The distributed nature of algorithms 1-3 also present the ability of trivially introducing additional mobile robots into the network. Each robot can implement algorithms 2, and 3 with absolutely no changes thereby accelerating localization of the sensor network to an even fine granularity.



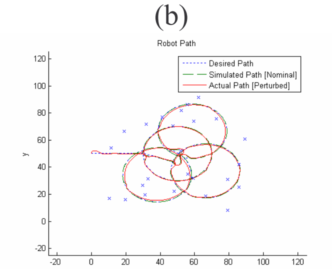
(a)



(b)



(c)



(d)

Figure 25: Adaptive localization paths with no initial sweep.

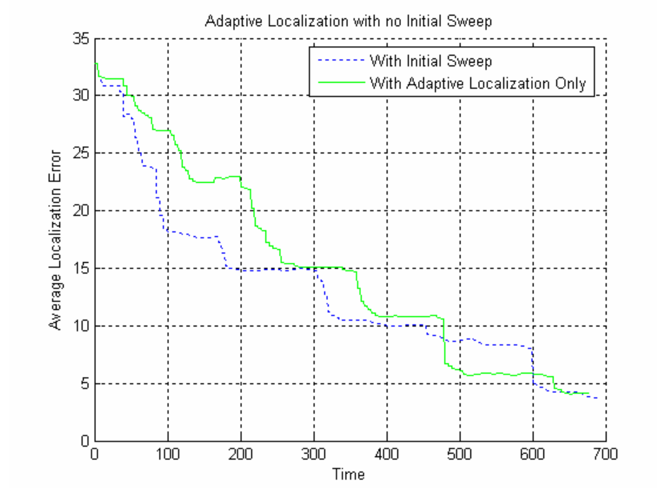


Figure 26: Adaptive localization error with no initial sweep.

VI. Conclusion

Rigorous mathematical algorithms for adaptive simultaneous localization of the static unattended ground sensors and the mobile robot have been demonstrated. The first algorithm localizes the static sensors and the second algorithm localizes the mobile robot. These algorithms together allow simultaneous localization of the static sensor and the mobile robot. A third adaptive localization algorithm ensures that the region of the deployment area with the largest uncertainty is localized with minimal robot movement.

Future work will involve experimental validations of the proposed algorithms using the Automation & Robotics Research Institue's (ARRI) WSN testbed and the mobile robot developed by the authors in [24]. A communication link estimation model as discussed in [40] may need to be added to the Kalman filter to reliably route NAV-REQ and LOC-REQ packets in an efficient manner in the presence of lossy communication links.

References

[1] N. Bulusu, J. Heidemann and D. Estrin, "Gps-less low cost outdoor localization for very small devices," IEEE Personal Communications Magazine, vol. 7, pp. 28-34, 2005.

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

[3] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, Global Positioning System : Theory and Practice, New York: Springer, 2004.

[4] R.L. Moses, D. Krishnamurthy and R.M. Patterson, "A self-localization method for wireless sensor networks," EURASIP Journal on Applied Signal Processing, 2002.

[5] N. Bulusu, J. Heidemann and D. Estrin, "Adaptive Beacon Placement," ICDCS '01: Proceedings of the the 21st International Conference on Distributed Computing Systems, pp. 489, 2001.

[6] C. Savarese, J.M. Rabaey and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001.

[7] N. Patwari, J.N. Ash, S. Kyperountas, A.O. Hero III, R.L. Moses and N.S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," Signal Processing Magazine, IEEE, vol. 22, pp. 54-69, July 2005.

[8] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proceedings of the IEEE Infocom 2000,* Tel-Aviv, Israel, pp. 775-784, Mar. 2000

[9] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, "The cricket location-support system," *Sixth Annual ACM International Conference on Mobile Computing and Networking (MOBICOM),* Boston, Massachusetts, August 2000.

[10] A. Savvides, C. Han and M.B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking,* Rome, Italy, pp. 166-179, 2001.

[11] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," *Proceedings of INFOCOM,* San Francisco, CA, 2003.

[12] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in *Proceedings of the 10th Annual International Converence on Mobile Computing and Networking,* September 2004, pp. 45-57.

[13] C. Taylor, A. Rahimi, J. Bachrach and H. Shrobe, "Simultaneous Localization and Tracing in an Ad Hoc Sensor Network," *Information Procession in Sensor Networks,* 2005.

[14] A.M. Brooks, S. Williams and A. Makarenko, "Automatic Online Localization of Nodes in an Active Sensor Network," International Conference on Robotics and Automation, vol. 5, pp. 4821-4826, 2004.

[15] S. Shenoy and J. Tan, "Simultaneous Localization and Mobile Robot Navigation in a Hybrid Sensor Network," IEEE/RSJ International Conference on Intelligent Robots and Systems, August 2005.

[16] A. Galstyan, B. Krishnamachari, K. Lerman and S. Pattem, "Distributed online localization in sensor networks using a moving target," IPSN'04: Proceedings of the Third International Symposium on Information Processing in Sensor Networks, pp. 61-70, 2004.

[17] V. Cevher and J.H. McClellan, "Sensor array calibration via tracking with the extended Kalman filter," IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2817-2820, 2001.

[18] P.N. Pathirana, N. Bulusu, A.V. Savkin and S. Jha, "Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks," *IEEE Transactions on Mobile Computing,* vol. 4, pp. 285-296, May/June 2005.

[19] E. Olson, J. Leonard and S. Teller, "Robust Range-Only Beacon Localization," in *IEEE/OES Autonomous Underwater Vehicles,* 2004, pp. 66-75.

[20] W. Xiao, J.K. Wu and L. Xie, "Sensor scheduling for target tracking in networks of active sensors," in *IEEE International Workshop on Sensor Networks and Applications,* 2005.

[21] Dissanayake, Gamini M. W. M., P. Newman, S. Clark and H.F. Durrant-Whyte, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," vol. 17, pp. 229-241, June 2001 2001.

[22] J.W. Fenwick, P.M. Newman and J.J. Leonard, "Cooperative Concurrent Mapping and Localization," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation,* May 2002, pp. 1810-1817.

[23] D.O. Popa, A.C. Sanderson, R.J. Komerska, S.S. Mupparapu, D.R. Blidberg and S.G. Chappel, "Adaptive sampling algorithms for multiple autonomous underwater vehicles," in *Autonomous Underwater Vehicles,* 2004, pp. 108-118.

[24] D.O. Popa, K. Sreenath and F.L. Lewis, "Robotic Deployment for Environmental Sampling Applications," in *International Conference on Control and Automation,* 2005, pp. 197-202.

[25] P.S. Maybeck, Stochastic models, estimation, and control, New York: Academic Press, 1979.

[26] W.S. Levine Ed., The Control Handbook, New York: CRC Press, 1996.

[27] S.I. Roumeliotis and G.A. Bekey, "An extended Kalman filter for frequent local and infrequent global sensor data fusion," in *SPIE International*

Symposium on Intelligent Systems and Advanced Manufacturing, pp. 11-22.

[28] E. Kiriy and M. Buehler, "Three-state Extended Kalman Filter for Mobile Robot Localization," McGill University., Montreal, Canada, Tech. Rep. TR-CIM 05.06, April, 2002.

[29] A. Kelly, "A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles," Carnegie Mellon University., Pittsburgh, PA, Tech. Rep. CMU-RI-TR-94-19, May, 1994.

[30] F.L. Lewis, Optimal Estimation, New York: John Wiley & Sons, 1986.

[31] D. Fox, W. Burgard, F. Dellaert and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99),* July, 1999.

[32] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation,* vol. 12, pp. 869-880, 1996.

[33] E.W. Weisstein, Frobenius Norm. From MathWorld--A Wolfram Web Resource. Available: http://mathworld.wolfram.com/FrobeniusNorm.html.

[34] G. Zhou, T. He, S. Krishnamurthy and J.A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," in *International Conference on Mobile Systems, Applications and Services,* 2004, pp. 125-138.

[35] G. Zhou, T. He, S. Krishnamurthy and J.A. Stankovic, "Models and Solutions for Radio Irregularity in Wireless Sensor Networks," *ACM Transactions on Sensor Networks,* vol. 2, pp. 221-262, May 2006.

[36] G. Lu, N. Sadagopan, B. Krishnamachari and A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communication Societies,* March 2005, pp. 2470-2481.

[37] L. Wang and Y. Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," *Mobile Networks and Applications,* vol. 11, pp. 723-740, October 2006.

[38] W. Ye, J. Heidemann and E. Deborah, "An Energy-Efficient MAC protocol for Wireless Sensor Networks," in *Proceedings of the IEEE Infocom,* 2002, pp. 1567-1576.

[39] T.V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems,* 2003, pp. 171-180.

[40] A. Woo, T. Tong and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems,* 2003, pp. 14-27.