Node Localization Using Mobile Robots in Delay-Tolerant Sensor Networks

Pubudu N. Pathirana, Nirupama Bulusu, *Member*, *IEEE*,
Andrey V. Savkin, *Senior Member*, *IEEE*, and Sanjay Jha, *Member*, *IEEE*

Abstract—We present a novel scheme for node localization in a Delay-Tolerant Sensor Network (DTN). In a DTN, sensor devices are often organized in network clusters that may be mutually disconnected. Some mobile robots may be used to collect data from the network clusters. The key idea in our scheme is to use this robot to perform location estimation for the sensor nodes it passes based on the signal strength 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 a Robust Extended Kalman Filter (REKF)-based state estimator to solve the localization. Compared to the standard extended Kalman filter, REKF is computationally efficient and also more robust. Finally, we have implemented our localization scheme on a hybrid sensor network test bed and show that it can achieve node localization accuracy within 1m in a large indoor setting.

Index Terms—Localization, delay-tolerant sensor networks, Robust Extended Kalman Filter, mobile robot, mobility.

1 Introduction

Recent years have witnessed a boom in sensor networks research [1], [2] and commercial activities [3]. This has been motivated by the wide range of potential applications from environmental monitoring to condition-based maintenance of aircraft. Sensor networks are frequently envisioned to exist at large scale and characterized by extremely limited end-node power, memory, and processing capability.

Several large scale environmental monitoring applications do not require a fully connected, uniformly distributed sensor network nor do they require real-time sensor data. Scientific analysis is based on sensor data collected over a longer period of time. For example, monitoring cane toads in the 200,000 sq km Kakadu National Park of Australia, long-term coastline monitoring, etc.

For such applications, the concept of a *delay-tolerant sensor network (DTN)* was first proposed by Fall [4]. A DTN would typically be deployed to monitor an environment over a long period of time and characterized by noninteractive sensor data traffic. Sensors are randomly scattered and organized into one or more clusters that may be disconnected from each other. Each cluster has a cluster head. Sensor information is typically aggregated at the cluster heads, which tend to have more resources and are responsible for communicating data to outside world. Wireless mobile robots (e.g., robomote [5]), unmanned aerial vehicles, can roam around the network to collect data from cluster heads or to dynamically reprogram or reconfigure the sensors. Examples of DTNs in existence are Sammi [6], Zebranet [7], and DataMules [8].

This paper revisits the problem of *node localization*, i.e., estimating sensor node positions for a delay-tolerant sensor network. A DTN has several distinguishing characteristics which motivate alternate approaches to node localization than those previously proposed. In a DTN, sensor nodes neither need to be localized in real time nor all at once. In this paper, we propose a novel localization scheme for DTNs using received signal strength (RSSI) measurements from each sensor device at a data gathering mobile robot. Our contributions are threefold:

- We motivate and propose a novel approach that allows one or more mobile robots to perform node localization in a DTN, 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.
- We develop a novel Robust Extended Kalman Filter (REKF)-based [9] state estimation algorithm for node localization in DTNs. Localization based on signal strength measurements is solved by treating it as online estimation in a nonlinear dynamic system (Section 3). Our model incorporates significant uncertainty and measurement errors and is computationally more efficient and robust in comparison to the extended Kalman filter implementation used to solve similar problems in cellular networks [10], [11] (Section 4).

- *P.N. Pathirana is with the School of Engineering and Technology, Deakin University, Pigdons Road, Geelong Vic 3217, Australia. E-mail: pubudu@deakin.edu.au.*
- *N. Bulusu is with National ICT Australia Limited, Bay 15, Australian Technology Park, Eveleigh 1431, Australia. E-mail: nbulusu@cse.unsw.edu.au.*
- *A.V. Savkin and S. Jha are with the School of Electrical Engineering and Telecommunications and the School of Computer Science and Engineering, respectively, University of New South Wales, Sydney 2052, Australia. E-mail: a.savkin@unsw.edu.au, sjha@cse.unsw.edu.au.*

Manuscript received 15 Mar. 2004; revised 10 May 2004; accepted 4 June 2004. For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0076-0304.

1536-1233/05/$20.00 © 2005 IEEE Published by the IEEE CS, CASS, ComSoc, IES & SPS

- We implement and validate our scheme on a novel hybrid sensor network testbed of motes, Stargates, and Lego Mindstorm robots (Section 5).

2 Related Work

In this section, we review research most relevant to our work: 1) delay-tolerant sensor networks and 2) sensor network localization.

2.1 Delay-tolerant Sensor Networks

Fall first proposed a Delay-tolerant Network architecture [4] for sensors deployed in mobile and extreme environments lacking an *always-on* infrastructure. These sensors are envisioned to monitor the environment over a long period of time. Herein, communication is based on an abstraction of message switching rather than packet switching. The abstraction of moderate-length messages (known as *bundles*) delivery for noninteractive traffic can provide benefits for network management because it allows the network path selection and scheduling functions to have a priori knowledge about the size and performance requirements of requested data transfers.

DTNs are already being used in practice. DataMules [8] uses a Mule that periodically visits sensor devices and collects information from these devices, in effect providing a message store-and-forward service that enables low-power sensor nodes to conserve power. The Sammi Network [6] is a community of Sammi people who are reindeer herders in Sweden and keep relocating their base. The Sammi communities do not have a wired or wireless communication infrastructure. Their relocation is controlled by a yearly cycle which depends on the natural behavior of the reindeer. In the Zebranet wildlife tracking system [7], wireless sensor nodes attached to animals collect location data and opportunistically report their histories when they come within radio range of base stations. While previous research has focused on communication abstractions, we are investigating the challenges and opportunities that arise from mobile data collecting elements in DTNs.

2.2 Sensor Network Localization

Localization is one of the most widely researched topics within the areas of sensor networks and robotics [12], [13], [14], [15]. Previous localization systems for sensor networks [13], [14] have been designed to *simultaneously* scale and *continuously* localize a large number of devices. To meet these requirements, devices usually compute their own location from their distance (or other measurement) made to nearby reference beacons. However, localization requirements for these sensor networks are different from delay-tolerant sensor networks.

In a DTN, nodes neither need to be localized concurrently nor continuously. We trade off node localization computation time for several other benefits. We can reduce the computational requirements for small sensor devices by instead using one or more mobile robots to compute the location of sensors. We can employ more sophisticated algorithms since processing is performed by the robot rather than sensor devices. We can also reduce the number of static location reference beacons required by exploiting the mobility of the robot. For instance, Eren et al. [16] estimate that, to uniquely localize a sensor network of n nodes in $O(\sqrt(\log n))$ steps, $O(\frac{n}{\log n})$ reference location beacons are required, using the iterative trilateration scheme proposed in [14]. To localize with just one beacon, $O(n)$ steps will be required.

In our scheme, assume that the mobile robot can localize $O(\log n)$ sensors in each step (a single Filter computation). This is not an unreasonable assumption to make since $\log n \leq 10$, even for a very large n. Using just one mobile robot, node localization can be achieved in $O(\frac{n}{\log n})$ steps. To localize in $O(\sqrt{\log n})$ steps, our scheme requires $O(\frac{n}{\log n\sqrt{\log n}})$ steps.

Cricket [13] and AHLoS [14] are ultrasound localization systems which can potentially provide superior accuracy to a radio-based localization scheme, such as the REKF proposed in this paper. However, compared to these systems, we require no additional hardware for extremely small sensor devices as our scheme leverages existing RF-communications capabilities.

Previous research has also investigated RSSI-based localization schemes for wireless networks, one of the first of which was RADAR [17]. One of the main drawbacks in RSSI-based localization schemes is the RSSI measurement noise caused by short-scale and medium-scale fading when both transmitter and receivers are stationary. In our scheme, we reduce the impact of fading to make RSSI-based localization more viable. Because the robot-receiver is mobile, over a period of time we can statistically eliminate the fading noise in RSSI measurements (not possible with a static transmitter-receiver pair).

Previously, Kalman filters and Bayesian filters have been applied to the localization problem (mainly in the context of robotics and cellular networks). In this paper, we propose using a Robust Extended Kalman Filter (REKF) as a state estimator in predicting sensor locations. These robust state estimation ideas emerged from the work of Savkin and Petersen [18]. It not only provides satisfactory results [19], but also eliminates requiring knowledge of measurement noise in the more commonly used standard Kalman filter implementation presented in [10]. It is significantly more computation and memory efficient than the more adaptive, but computationally complex and memory-intensive Bayesian filters, making it better suited to the sensor networks regime. In the next section, we describe in detail our node localization scheme using mobile robots in a delay-tolerant sensor network.

3 Localization Methodology

To solve node localization based on RSSI measurements at a mobile robot, we model it as an online estimation in a nonlinear dynamic system. In this section, we describe this system dynamic model and the nonlinear measurement model. We present the theoretical background for the

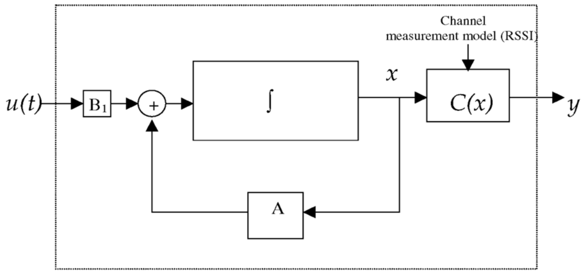


Fig. 1. Location estimation system.

Robust Extended Kalman filter used with this model in Appendix A.

3.1 System Dynamic Model

We use the terminology mobile robot for a mobile node fitted with a wireless base station. The sensors to be located are randomly distributed in an environment. The dynamic model for n sensors and the mobile robot can be given in two-dimensional cartesian coordinates as [20]:

$$\dot{x}(t) = Ax(t) + B_1u(t) + B_2w(t) \tag{1}$$

where

$$A = \begin{bmatrix} \Theta & & 0 \\ & \ddots & \\ 0 & & \Theta \end{bmatrix}, \quad -B_1 = \begin{bmatrix} \Phi \\ \vdots \\ \Phi \end{bmatrix}, \quad B_2 = \begin{bmatrix} \Phi & & 0 \\ & \ddots & \\ 0 & & \Phi \end{bmatrix},$$

$$\Theta = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \Phi = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The dynamic state vector $x(t) = [x_1(t) \dots x_i(t) \; \dots x_n(t)]'$ with $x_i(t) = \left[\mathrm{X}_i(t)\; \mathrm{Y}_i(t)\; \dot{\mathrm{X}}_i(t)\; \dot{\mathrm{Y}}_i(t)\right]'$, where $i \in [1 \dots n]$, $\mathrm{X}_i(t)$, and $\mathrm{Y}_i(t)$ represent the position of the ith sensor ($\mathrm{Sensor_i}$) with respect to the mobile robot at time t, and their first order derivatives, $\dot{\mathrm{X}}(t)$ and $\dot{\mathrm{Y}}(t)$, represent the relative speed along the X and Y directions. In other words, if $x_c(t) = [\mathbf{x}_c(t)\; \mathbf{y}_c(t)\; \dot{\mathbf{x}}_c(t)\; \dot{\mathbf{y}}_c(t)]'$ represents the absolute state (position and velocity in order in the X and Y direction, respectively) of the mobile robot and $x_s^i(t) = \left[\mathbf{x}_s^i(t)\; \mathbf{y}_s^i(t)\; \dot{\mathbf{x}}_s^i(t)\; \dot{\mathbf{y}}_s^i(t)\right]'$ denotes the absolute state of the $\mathrm{Sensor_i}$ in the same order, then $x_i(t) \triangleq x_c(t) - x_s^i(t)$. Furthermore, let $u(t)$ denote the two-dimensional driving/acceleration command of the mobile robot from the respective accelerometer readings and $w(t)$ denote the unknown two-dimensional driving/acceleration command of the sensor if moving. Although it can be generalized for the moving sensor case as most applications rely on stationary sensors, here we consider the sensors as stationary and set $w(t) = 0$. This system can be represented in graphical form in the form of an input $(u(t))$ and measurement (y) system, as in Fig. 1. We omitted B_2 as we only consider the case of stationary sensors. The basic idea in such a system is to estimate state x from measurement y. In the localization problem, as the sensor locations are unknown, we assume an arbitrary location (0, 0). We show that this assumed state converges to the actual state and, hence, the unknown sensor location can be estimated (as the position/state of the mobile robot is known) within the prescribed time frame.

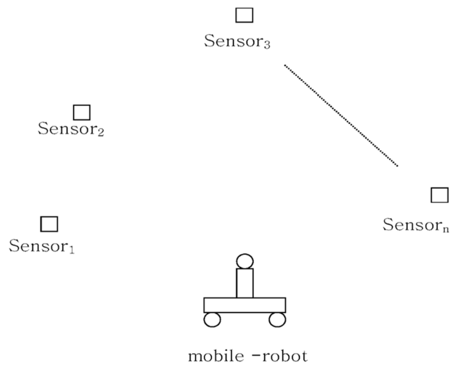


Fig. 2. Network geometry.

3.2 RSSI Measurement Model

In wireless networks, the distance between two communicating entities is observable using the forward link RSSI (received signal strength indication) of the receiver. When multiple transmitters are present, the data association is unambiguous, i.e., which measurement comes from which transmitter can be precisely determined simply by examining the source (transmitter) identifier in the data packet.

Measured in decibels at the mobile robot for our case, RSSI can be modeled as a two-fold effect: due to *path loss* and due to *shadow fading* [10]. Fast fading is neglected assuming that a low-pass filter is used to attenuate Rayleigh or Rician fade. Denoting the ith sensor as $\mathrm{Sensor_i}$ (Fig. 2), the RSSI from the $\mathrm{Sensor_i}$, $p_i(t)$ can be formulated as [21]

$$p_i(t) = p_{oi} - 10\varepsilon \log d_i(t) + v_i(t), \tag{2}$$

where p_{oi} is a constant determined by the transmitted power, wavelength, and antenna gain of the mobile robot. ε is a slope index (typically 2 for highways and 4 for microcells in the city) and $v_i(t)$ is the logarithm of the shadowing component, which is considered as an uncertainty in the measurement. $d_i(t)$ represents the distance between the mobile robot and $\mathrm{Sensor_i}$, which can be further expressed in terms of the position of the ith sensor with respect to the location mobile robot, i.e., $(\mathrm{X}_i(t), \mathrm{Y}_i(t))$

$$d_i(t) = \left(\mathrm{X}_i(t)^2 + \mathrm{Y}_i(t)^2\right)^{1/2} \tag{3}$$

For sensors within a network cluster, we use measurements at a single mobile robot, as opposed to multiple ones [10]. The observation vector,

$$y(t) = \begin{bmatrix} p_1(t) \\ \vdots \\ p_n(t) \end{bmatrix}, \tag{4}$$

is sampled progressively as the mobile robot moves in the coverage area. The measurement equation for the

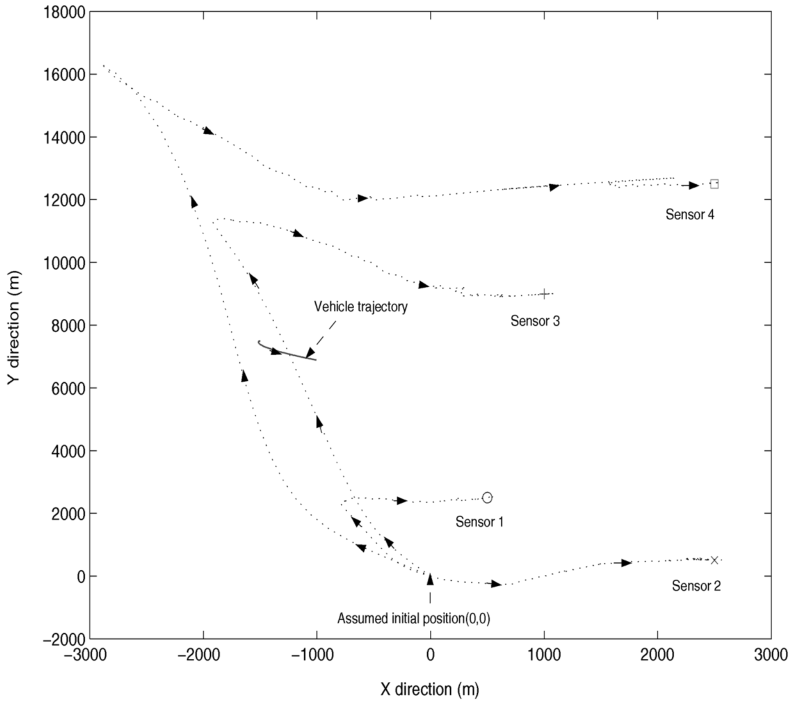


Fig. 3. Location estimation trajectories converging to the actual sensor locations.

measurements made by the mobile robot for the n number of sensors are in the form of

$$y(t) = C(x(t)) + v(t), \tag{5}$$

where $v(t) = [v_1(t) \cdots v_n(t)]'$ with

$$C(x(t)) = \begin{bmatrix} p_{oi} - 10\varepsilon \log\left(\mathrm{X}_1(t)^2 + \mathrm{Y}_1(t)^2\right) \\ \vdots \\ p_{oi} - 10\varepsilon \log\left(\mathrm{X}_n(t)^2 + \mathrm{Y}_n(t)^2\right) \end{bmatrix}. \tag{6}$$

We provide a brief intuitive explanation of REKF here (see the Appendix for a detailed theoretical background). We use the state space model (a representation of the dynamic system consisting of the mobile robot and the n sensors using a set of differential equations derived from simple kinematic equations). Our dynamic system considers two noise inputs: 1) measurement noise (this is standard with any measurement), v in $y = C(x) + v$, and 2) w-*acceleration*, which is also considered noise as it is unknown. In this application, the initial condition errors are quite significant as no knowledge is available regarding the sensor locations. This issue is directly addressed as the proposing algorithm is inherently robust against estimation errors of the initial condition (see (21) in the Appendix). The two noise inputs and the initial estimation errors have to satisfy the IQC equation (see (21) in the Appendix). If there exists a solution to the Ricati equation (see Section A.2 in the Appendix), then the IQC, presented in a suitable form by (21), is satisfied and the states can be estimated from the measurements using (23), which is a robust version of the Extended Kalman Filter (REKF).

In the application of REKF in a delay-tolerant network, the ith system (the mobile robot and the $\mathrm{Sensor_i}$), during a corresponding time interval, is represented by the nonlinear, uncertain system in (16), together with the following Integral Quadratic Constraint (IQC) (from (21)) :

$$\begin{aligned} &(x(0) - x_0)' N_i (x(0) - x_0) \\ &+ \frac{1}{2}\int_0^s \left(w(t)' Q_i(t) w(t)\right) + v(t)' R_i(t) v(t) dt \\ &\leq d + \frac{1}{2}\int_0^s z(t)' z(t) dt. \end{aligned} \tag{7}$$

Here, $Q_i > 0$, $R_i > 0$, and $N_i > 0$ with $i \in \{1, 2, 3\}$ are the weighting matrices for each system i. The initial state (x_0) is the estimated state of respective systems at startup. It is essentially derived from the terminal state of the previous system, together with other data available in the network (i.e., robot position and speed) to be used as the initial state for the next system taking over the tracking. With an uncertainty relationship of the form of (7), the inherent measurement noise (see (5)), the unknown mobile robot acceleration, and the uncertainty in the initial condition are considered as bounded deterministic uncertain inputs. In particular, the measurement equation with the standard norm bounded uncertainty can be written as (5):

$$y = C(x) + \delta C(x) + v_0, \tag{8}$$

where $|\delta| \leq \xi$, with ξ being a constant indicating the upper bound of the norm bounded portion of the noise. By choosing $z = \xi C(x)$ and $\nu = \delta C(x)$,

$$\int_0^T |\nu| dt \leq \int_0^T z' z dt. \tag{9}$$

TABLE 1
Simulation Parameters for Scenario 1

Parameter	Value	Comments
p_{oi}	20w	Base station transmission power
N	diag{10,10,10,1, 70,25,2,5, 100,1,20,1, 40,40,20,1}	Weighting on the Initial viscosity solution
Q	$\text{diag}\{2, 2, 11, 1, 1, 3, 1, 2\}$	Weighting on the uncertainty in the vehicle driving command
R	$\text{diag}\{2 \times 10^3, 1.7 \times 10^3,\}$ $10 \times 10^3, 51 \times 10^3$	Weighting on the measurement noise
T	$5mins$	Simulation time
A_{max}	$50m/s^{-2}$	weighting on u(t)
Ts	2s	Sampling interval
$x_s^1(0)$	[500m 2500m 0 0]'	1^{st} sensor initial state
$x_s^2(0)$	[2500m 500m 0 0]'	2^{nd} sensor initial state
$x_s^3(0)$	[1000m 9000m 0 0]'	3^{rd} sensor initial state
$x_s^3(0)$	[2500m 12500m 0 0]'	4^{th} sensor initial state
$x_c(0)$	[2500m -2500m $20ms^{-1}$ $10ms^{-1}$]'	$mobile - robot$ initial state

Considering v_0 and the corresponding uncertainty in w as w_0 satisfying the bound

$$\Phi(x(0)) + \int_0^T \left[w_0(t)'Qw_0(t) + v_0(t)'Rv_0(t)\right]dt \leq d, \qquad (10)$$

it is clear that this uncertain system leads to the satisfaction of condition in (17) and, hence, (21) (see [9]). This more realistic approach removes any noise model assumptions in algorithm development and guarantees the robustness.

3.3 Robust versus Optimal State Estimation

REKF tends to increase the robustness of the state estimation process and reduce the chance that a small deviation from the Gaussian process in the system noise causes a significant negative impact on the solution. However, we lose optimality and our solution will be just suboptimal. To explain the connection between REKF and the standard extended Kalman Filter, consider the system (16) with

$$K(x, u) = \nu K_0(x, u), \qquad (11)$$

where $K_0(x, u)$ is some bounded function and $\nu > 0$ is a parameter. Then, the REKF estimate $\tilde{x}(t)$ for the system (11), (16), and (21), as defined by (23) and (24), converges to $\tilde{x}^0(t)$, as ν tends to 0. Here, $\tilde{x}^0(t)$ is the extended Kalman state estimate for the system (16) with the Gaussian noise $\left[\, w(t)' \quad v(t)' \,\right]$ satisfying

$$E\left\{\begin{bmatrix} w(t) \\ v(t) \end{bmatrix}\left[\, w(t)' \quad v(t)' \,\right]\right\} = \begin{bmatrix} Q(t) & 0 \\ 0 & R(t) \end{bmatrix}.$$

The parameter ν in (11) describes the uncertainty in the system and measurement noise. For small ν, our robust state estimate becomes close to the Kalman state estimate with Gaussian noise. For larger ν, we achieve more robustness, but less optimality. We show via simulation that, for larger uncertainty (which is quite realistic), our robust filter still performs well, whereas the standard extended Kalman estimate diverges.

4 Simulations

To examine the performance of the Robust Extended Kalman Filter for a sensor network, we simulate a mobile robot equipped with a radio transceiver moving in the sensor coverage area. We assume the network knows the acceleration and position of the mobile robot via GPS and accelerometer readings, but has no information about the sensors. We simulate two scenarios—large sensors and small sensors.

4.1 Large Sensors

In Scenario 1, we simulate large sensors scattered over a wide area. We expect sensors to be low cost and equipped with modest transmitters. To model this, we use a slow sampling rate of 2 seconds per sample. We simulate a mobile robot and four sensors. The algorithm can be scaled to as many sensors as required by appropriately increasing the number of mobile robots. Simulation parameters are listed in Table 1. The mobile robot measures the forward link signal from the four sensors and estimates the state of

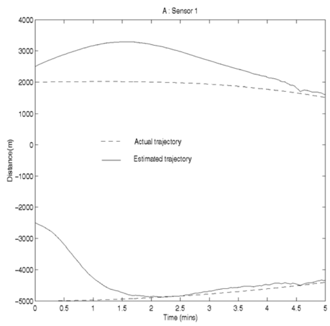


Fig. 4. Estimation convergence for Sensor 1.

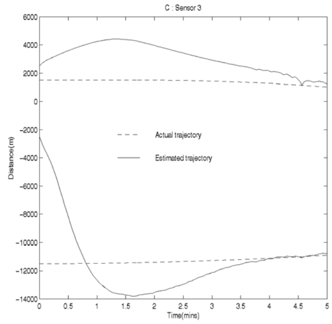


Fig. 6. Estimation convergence for Sensor 3.

the system from an arbitrary initial estimate (zero). Fig. 1 shows how the estimated sensor location from an initial position converges to the actual sensor positions within the simulation time. Fig. 4, Fig. 5, Fig. 6, and Fig. 7 show the distance variation in the X and Y directions separately for each sensor, as well as the predicted distances approaching the actual distances. Fig. 9 shows that the extended Kalman filter cannot be used with large uncertain instances as it diverges.

4.2 Small Sensors

In Scenario 2, we use sensors with much less signal strength (600mW), as in [22], with higher sampling rate, as in commercially available systems. To demonstrate scalability, we increase both the time scale and the number of sensors in this scenario. The simulation parameters are given in Table 2. The arbitrary acceleration of the mobile robot is taken as $u(t) = A_{max}[-3\sin(0.2t) + \phi_1 \quad 0.9\cos(0.05t) + \phi_2]$.

In the dynamic system simulation, we choose the functions given in Table 2 for the arbitrary mobile robot acceleration (u), with ϕ_1 and ϕ_2 being uniform random distributions in the interval $[0 \ 0.1A_{max}]$. We consider uniformly distributed measurement noise in the interval $[0 \quad 0.01\|y(t)\|]$ with $y(t)$ being the noise-free measurement and $\xi = 0.05$. The equation for the state estimation and the corresponding Riccati Differential equation obtained from (23) and (24) are:

$$\begin{gathered}\dot{\tilde{x}}(t) = A\tilde{x}(t) + B_1 u_i(t) \\ +X^{-1}(t)[\beta^1(\tilde{x}(t))' R_i (y(t) - \beta(\tilde{x}(t))) \\ +\xi^2 \beta^1(\tilde{x}(t))' \beta^1(\tilde{x}(t))] \\ \tilde{x}(t) = x_0\end{gathered} \tag{12}$$

and

$$\begin{gathered}\dot{X} + A'X + XA + XB_2 Q_i^{-1} B_2' X - \\ \beta^1 \tilde{x}(t)' R_i \beta^1 \tilde{x}(t) + \xi^2 \beta^1(\tilde{x}(t))' \beta^1(\tilde{x}(t)) = 0 \\ X(0) = N,\end{gathered} \tag{13}$$

where

$$\beta(x) = C(x(t), \tag{14}$$

as shown in (6). Also, here,

$$\beta^1(x) = \nabla_x \beta(x), \tag{15}$$

where x_0 is $x(0)$, the relative initial dynamic state of the system. In the second scenario, 10 sensors with lesser signal strength are used with the mobile robot. Fig. 8a plots the estimated trajectory approaching each respective sensor location from an initial estimate of each sensor location of (0, 0). Fig. 8b plots the percentage error in localizing each sensor with respect to the initial estimation error.

5 Implementation and Experimental Results

We have implemented our REKF-based localization scheme to verify its computational efficiency and estimation accuracy in a real environment. We now describe this implementation and report on preliminary experimental results.

5.1 Hybrid Sensor Platform

Our hybrid platform consists of three devices: 1) motes, 2) Stargates, and 3) Lego Mindstorm robots. They differ in processing, memory, battery, and mobility capabilities, as well as in their operating systems software.

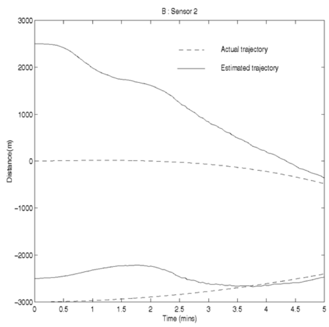


Fig. 5. Estimation convergence for Sensor 2.

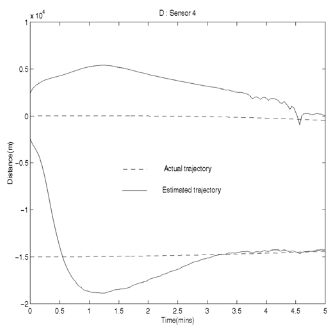
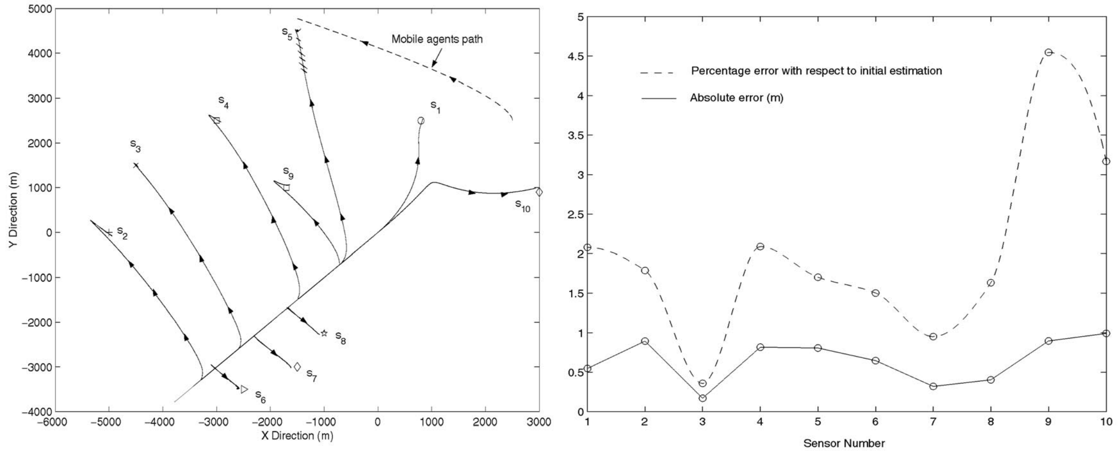
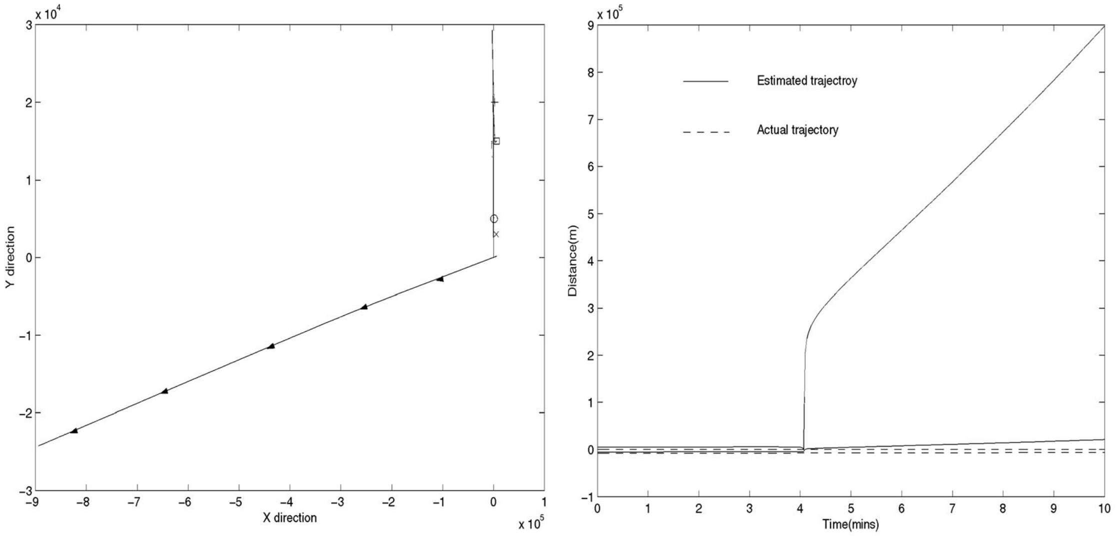


Fig. 7. Estimation convergence for Sensor 4.



(a) (b)

Fig. 8. (a) Estimation path for each sensor (Scenario 2). (b) Percentage error and the absolute error from the initial estimation.



(a) (b)

Fig. 9. (a) Using the standard Kalman filter as the state estimator. (b) Divergence of state estimation when using the standard Kalman filter.

TABLE 2
Stargate and MICA2 Comparison

Platform	Core	Processor	Data Path	MHz	Memory
MICA2	Atmega 128L4	RISC	8 bits	4 MHz	128 KB program flash, 4 KB data
Stargate	Intel PXA255 Xscale	RISC	32 bits	400 MHz	32 MB flash, 64 MB SDRAM

5.1.1 Motes

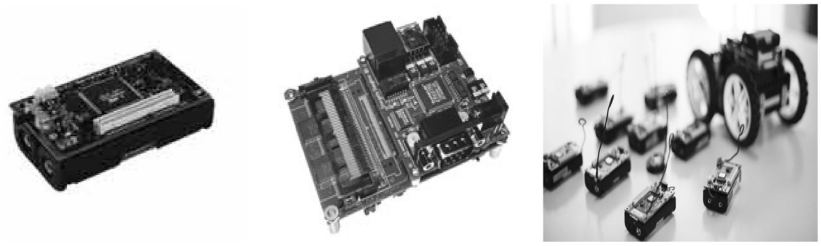
Motes [23], shown in Fig. 10, are our resource impoverished devices that run the TinyOS event-driven operating system. The Mica2 mote sensors deployed in our experiment use the CC1000 radio from ChipCon, which provides an analog RSSI measurement that can be connected to an analog to digital converter (ADC) to produce digital signals. These RSSI measurements can be used for localization.

5.1.2 Stargate

Stargate [24], shown in Fig. 10, is a resource-rich node that provides more capabilities than the MICA motes. It is a powerful Linux-based single board computer with an Intel 400MHz X-Scale processor (PXA255), Compact Flash, PCMCIA, Ethernet, USB Host, 64 MB SDRAM, and an additional interface to communicate with a mote. We use a Stargate as the computational substrate for the mobile robot. The Stargate runs the Robust Extended Kalman Filter which is implemented in Java. RSSI readings are measured for the mote interfaced with the Stargate communicating with other sensors.

5.1.3 Lego MindStorm: Mobility

We use the popular Lego MindStorm [25] platform, shown in Fig. 10, to emulate a mobile robot. It is a programmable,



(a) (b) (c)

Fig. 10. Hardware components of the hybrid test bed. (a) Mica2 mote. (b) Stargate. (c) Lego Mindstorm robot.

nonmaneuvring robot that is constructed by connecting small Legos together. A Stargate is mounted onto the Lego. An Infra Red tower is used to program the RCX box, which controls the Lego MindStorm. At the core of the RCX is a Hitachi H8 microcontroller with 32K external RAM. The microcontroller is used to control three motors, three sensors, and an infrared serial communications port. Both the driver and firmware accept and execute commands from the PC through the IR communications port. To calculate the velocity, the Lego MindStorm is programmed to move at a constant speed (selected from eight power options) in a straight line.

5.2 Experimental Results

We placed the motes in the topology shown in Fig. 11 and programmed the Lego Mindstorm robot to move in a straight line. We measured the distance traveled and time elapsed to accurately deduce the velocity of a mobile robot (sampling rate = 0.3 seconds). The values of the weighting matrices are tuned with simulations using the modeled system parameters.

5.2.1 Computational Efficiency

To characterize computational efficiency, we measured the REKF computation time (for 10 sensors) as a function of the number of RSSI samples for the following cases:

- Java implementation (Sun's JVM) on a Pentium IV 3GHz machine,
- Matlab implementation on a Pentium IV 3GHz machine, and
- Java implementation, with Open-Wonka on a Stargate 400MHz machine.

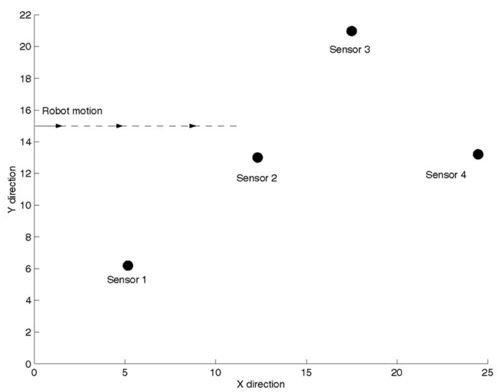


Fig. 11. Indoor sensor distribution and robot navigation topology.

The computation time in milliseconds is shown in Table 3. The performance of Matlab and Java running on a Pentium IV 3 GHz does not differ much (in the same order). However, for a Stargate, the performance degrades significantly with the number of samples. The Stargate has very limited memory (64 MB SDRAM), which makes Open-Wonka's garbage collector inefficient. To improve the performance on the Stargate, we can 1) break down the computation on Stargate to a smaller subsets of samples or 2) implement REKF in C instead of Java.

5.2.2 Estimation Accuracy

To evaluate estimation accuracy, we report on an experiment with four sensors since the visualization of estimation convergence is clearer with a smaller number of sensors. The four sensors are positioned at $(6.1, 6)m$, $(12.2, 13.6)m$, $(18.3, 21.2)m$, and $(24.4, 13.6)m$. The mobile vehicle (robot) is initially located at $(0, 15.2)m$ and moves with a velocity of $2.52m/min$. For our indoor implementation, with the collected data and modeling the RSSI, we use $p_{oi} = 160mw$ and $\varepsilon = 3$ (see (2)).

Fig. 12 shows the localization of the four sensors with approximate error of $1m$. The results are very close to the real positions. The Lego Mindstorm is a nonmaneuvering robot (constant velocity). By incorporating vehicle maneuvers, further improvements to the localization error can obviously be made.

6 Algorithmic Improvements and Future Work

Experimental evaluation revealed the importance of finding the right parameters for the weighting matrices, which model the weighting between different measurement noises. Our next step is to implement automatic tuning of these parameters through machine learning techniques to enhance the usability of our localization scheme.

The algorithm stores all sensor states during the experiment period in order to validate the convergence of estimations. To improve computational efficiency in a production system, where only the final estimation is needed, only two states for each sensor need to be stored (instead of nearly 500).

In general, the use of mobile robots in Delay-Tolerant Sensor Networks opens up a number of other interesting research possibilities. Once sensor localization has been performed, it is possible to create a topological map (or a

TABLE 3
Computation Times for REKF Implementations

Number of Samples	Pentium-IV (Java) (seconds)	Pentium-IV (MATLAB) (seconds)	STARGATE (Java) (seconds)
90	-	-	115
2000	4	70	-
8000	15	27	-
16000	27	56	-

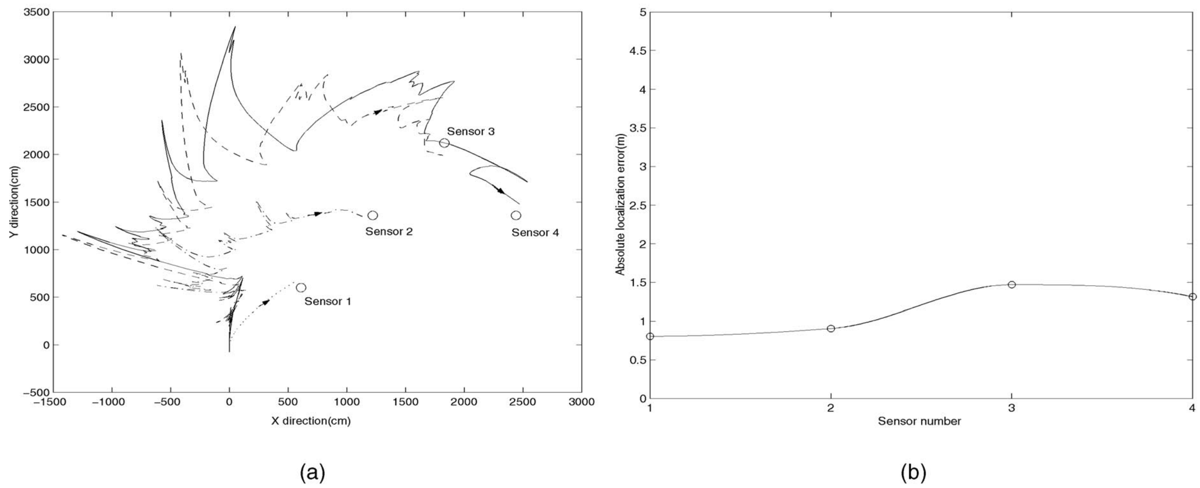


Fig. 12. (a) Sensor localization, and (b) localization error for the real system.

path profile) that can be optimized so that the data collecting mobile robot can follow this path for data collection in the shortest possible time or to meet storage and power requirements. Mobile robots could act as relays between disconnected portions of the network, thereby forming a relay network. The trajectory of a mobile robot can be dynamically recalculated so that a mobile robot can slow down whenever it needs to download a lot of information. We intend to explore these aspects in future work.

7 Conclusions

In this paper, we have provided a scheme for node localization using mobile robots in a delay-tolerant sensor network (DTN). To the best of our knowledge, no other study has been done for such a network. DTNs are commonly deployed in long-term environmental monitoring applications. In a DTN, node localization does not need to happen in real time. Using one or more mobile robots to compute the location of sensors allows us to trade off the computational time for node localization for several other benefits. First, we can eliminate processing constraints for small sensor devices. We can employ more sophisticated algorithms since processing is performed by the robot rather than sensor devices. Second, we can reduce the number of static location reference beacons required by exploiting the mobility of the robot. Third, it makes RSSI-based localization more viable. Because the robot-receiver is mobile, over a period of time we can statistically eliminate the fading noise in RSSI measurements.

We proposed applying a Robust Extended Kalman Filter-based state estimator for node localization. It is computationally more efficient and robust to measurement noise than the more commonly used extended Kalman filter implementation. Real experiments in a large indoor area show that the localization accuracy is approximately 1m. This compares favorably with the previously proposed RSSI localization schemes in an indoor setting [17] (accuracies within 3m), as well as with finer-grained acoustic time-of-flight localization schemes [13], [14] (accuracies vary 10cm-25cm). Now that we have validated our ideas through simulation, implementation, and experiment, we are working on further localization scheme refinements and on other mobile robot uses in delay-tolerant sensor networks.

Appendix A

We consider a nonlinear uncertain system of the form

$$\begin{aligned} \dot{x} &= A(x,u) \quad + B_2 w \\ z &= K(x,u) \\ y &= C(x) \qquad + v \end{aligned} \tag{16}$$

TABLE 4
Simulation Parameters for Scenario 2

Parameter	Value	Comments
p_{oi}	600mw	Base station transmission power
N	$\text{diag}\{10^{-2}, 10^{-2}, 10^{-2}, 10^{-2},$ $10^{-8}, 10^{-8}, 10^{-8}, 10^{-8},$ 5,5,5,5, 5,5,5,5, 5,5,5,5, $10^{-4}, 10^{-4}, 10^{-4}, 10^{-4},$ $0.07, 0.07, 0.07, 0.07,$ $0.07, 0.07, 0.07, 0.07,$ $0.03, 0.03, 0.03, 0.03,$ $10^{-4}, 10^{-4}, 10^{-4}, 10^{-4}\}$	Weighting on the Initial viscosity solution
Q	$\text{diag}\{10^4, 10^4, 10^8, 10^8,$ $5 \times 10^3, 5 \times 10^3,$ $5 \times 10^3, 5 \times 10^3,$ $5 \times 10^3, 5 \times 10^3,$ $5 \times 10^{11}, 5 \times 10^{11},$ $10^{11}, 10^{11}, 10^{11}, 10^{11},$ $10^{14}, 10^{14}, 10^9, 10^9, \}$	Weighting on the uncertainty in the user driving command
R	$\text{diag}\{100, 3, 2 \times 10^9$ $2 \times 10^9, 2 \times 10^9$ $2 \times 10^9, 6 \times 10^5$ $64 \times 10^7, 64 \times 10^7$ $5 \times 10^5, 65\}$	Weighting on the measurement noise
T	$10mins$	Simulation time
A_{max}	$50m/s^{-2}$	weighting on u(t)
Ts	0.05s	Sampling interval
$x_s^1(0)$	[800m 2500m 0 0]'	1^{st} sensor initial state
$x_s^2(0)$	[-5000m 0 0 0]'	2^{nd} sensor initial state
$x_s^3(0)$	[-4500m 1500m 0 0]'	3^{rd} sensor initial state
$x_s^4(0)$	[-3000m 2500m 0 0]'	4^{th} sensor initial state
$x_s^5(0)$	[-1500m 4500m 0 0]'	5^{th} sensor initial state
$x_s^6(0)$	[-2500m -3500m 0 0]'	6^{th} sensor initial state
$x_s^7(0)$	[-1500m -3000m 0 0]'	7^{th} sensor initial state
$x_s^8(0)$	[-1000m -2250m 0 0]'	8^{th} sensor initial state
$x_s^9(0)$	[-1700m 1000m 0 0]'	9^{th} sensor initial state
$x_s^{10}(0)$	[3000m 900m 0 0]'	10^{th} sensor initial state
$x_c(0)$	[2500m 2500m $2ms^{-1}$ $10ms^{-1}$]'	$mobile-robot$ initial state

as a general form of the system given by (1) with measurement equation in the form of (5) and defined on the finite time interval $[0, s]$. Here, $x(t) \in \mathbb{R}^n$ denotes the *state* of the system, while $y(t) \in \mathbb{R}^l$ is the *measured output* and $z(t) \in \mathbb{R}^q$ is the *uncertainty output*. The uncertainty inputs are $w(t) \in \mathbb{R}^p$ and $v(t) \in \mathbb{R}^l$. Also, $u(t) \in \mathbb{R}^m$ is the known *control input*. We assume that all of the functions appearing in (16) are with continuous and bounded partial derivatives. Additionally, we assume that $K(x, u)$ is bounded. This was assumed to simplify the mathematical derivations and can be removed in practice [9], [26]. The matrix B_2 is assumed to be independent of x and is of full rank.

The uncertainty in the system is defined by the following *nonlinear integral constraint* [9], [27]:

$$\Phi(x(0)) + \int_0^s L_1(w(t), v(t))dt \le d + \int_0^s L_2(z(t))dt, \tag{17}$$

where $d \ge 0$ is a positive real number. Here, Φ, L_1, and L_2 are bounded, nonnegative functions with continuous partial derivatives, satisfying growth conditions of the type

$$\|\phi(x) - \phi(x^{'})\| \quad \le \quad \beta\Big(1 + \|x\| + \|x^{'}\|\Big)\|x - x^{'}\|, \tag{18}$$

where $\|\cdot\|$ is the Euclidian norm with $\beta > 0$ and $\phi = \Phi, L_1, L_2$. Uncertainty inputs $w(\cdot), v(\cdot)$ satisfying this condition are called *admissible uncertainties.* We consider the problem of characterizing the set of all possible states $\mathcal{X}_s$ of the system (16) at time $s \ge 0$, which are consistent with a given control input $u^0(\cdot)$ and a given output path $y^0(\cdot)$, i.e., $x \in \mathcal{X}_s$ if and only if there exist admissible uncertainties such that, if $u^0(t)$ is the control input and $x(\cdot)$ and $y(\cdot)$ are resulting trajectories, then $x(s) = x$ and $y(t) = y^0(t)$ for all $0 \le t \le s$.

A.1 The State Estimator

The state estimation set $\mathcal{X}_s$ is characterized in terms of level sets of the solution $V(x, s)$ of the PDE

$$\begin{aligned}\frac{\partial}{\partial t}V + \quad & \max_{w\in\mathbb{R}^m}\{\nabla_x V.(A(x,u^0) + B_2 w) \\ & -L_1(w, y^0 - C(x)) + L_2(K(x,u^0))\} = 0 \\ & V(\cdot, 0) = \Phi.\end{aligned} \tag{19}$$

The PDE (19) can be viewed as a filter, taking observations $u^0(t), y^0(t), 0 \le t \le s$ and producing the set $\mathcal{X}_s$ as a output. The state of this filter is the function $V(\cdot, s)$; thus, V is an information state for the state estimation problem.

Theorem 1. *Assume the uncertain system (16), (17) satisfies the assumptions given above. Then, the corresponding set of possible states is given by*

$$\mathcal{X}_s = \{x \in \mathbb{R}^n : V(x, s) \le d\}, \tag{20}$$

where $V(x,t)$ is the unique viscosity solution of (19) in $C(\mathbb{R}^n \times [0, s])$.

Proof. See [9]. □

A.2 A Robust Extended Kalman Filter

Here, we consider an approximation to the PDE (19), which leads to a Kalman filter-like characterization of the set $\mathcal{X}_s$. Petersen and Savkin in [9] presented this as a Extended Kalman filter version of the solution to the Set Value State Estimation problem for a linear plant with the uncertainty described by an Integral Quadratic Constraint (IQC). This IQC is also presented as a special case of (17). We consider the uncertain system described by (16) and an integral quadratic constraint of the form

$$\begin{aligned}& (x(0) - x_0)^{'} X_0 (x(0) - x_0) \\ +\tfrac{1}{2}\textstyle\int_0^s & \Big(w(t)^{'} Q(t) w(t)\Big) + v(t)^{'} R(t) v(t) dt \\ & \le d + \tfrac{1}{2}\textstyle\int_0^s z(t)^{'} z(t) dt,\end{aligned} \tag{21}$$

where $N > 0, Q > 0$ and $R > 0$. For the systems (16) and (21), the PDE (19) can be written as

$$\begin{aligned}\frac{\partial}{\partial t}V + \nabla_x V.A(x,u^0) + \tfrac{1}{2}\nabla_x V B_2 Q^{-1} B_2^{'} \nabla_x V^{'} \\ -\tfrac{1}{2}(y^0 - C(x))^{'} R (y^0 - C(x)) \\ +\tfrac{1}{2}K(x,u^0)^{'} K(x,u^0) = 0, \\ V(x,0) = (x - x_0)^{'} N (x - x_0).\end{aligned} \tag{22}$$

Considering a function $\hat{x}(t)$ defined as $\hat{x}(t) \triangleq \arg\min_x V(x,t)$, and the following equations (23), (24), and (25), define our approximate solution to the PDE (22):

$$\begin{aligned}\dot{\tilde{x}}(t) = \quad & A(\tilde{x}(t), u^0) \\ & +X^{-1}[\nabla_x C(\tilde{x}(t))^{'} R(y^0 - C(\tilde{x}(t))) \\ & +\nabla_x K(\tilde{x}(t), u^0)^{'} K(\tilde{x}(t), u^0)], \\ & \tilde{x}(t) = x_0.\end{aligned} \tag{23}$$

$X(t)$ is defined as the solution to the Riccati Differential Equation (RDE)

$$\begin{aligned}\dot{X} + \nabla_x A(\tilde{x}, u^0)^{'} X + X \nabla_x A(\tilde{x}, u^0) \\ +X B_2 Q^{-1} B_2^{'} X - \nabla_x C(\tilde{x})^{'} R \nabla_x C(\tilde{x}) \\ +\nabla_x K(\tilde{x}, u^0)^{'} \nabla_x K(\tilde{x}, u^0) = 0, \\ X(0) = N\end{aligned} \tag{24}$$

and

$$\begin{aligned}\phi(t) \triangleq \tfrac{1}{2}\textstyle\int_0^t [(y^0 - C(\tilde{x}))^{'} R (y^0 - C(\tilde{x})) \\ -K(\tilde{x}, u^0)^{'} K(\tilde{x}, u^0)] d\tau.\end{aligned} \tag{25}$$

The function $V(x,t)$ was approximated by a function of the form

$$\tilde{V}(x,t) = \frac{1}{2}(x - \tilde{x}(t))^{'} X(t) (x - \tilde{x}(t)) + \phi(t).$$

Hence, it follows from Theorem 1 that an approximate formula for the set $\mathcal{X}_s$ is given by

$$\tilde{\mathcal{X}}_s = \left\{x \in \mathbb{R}^n : \frac{1}{2}(x - \tilde{x}(s))^{'} X(s)(x - \tilde{x}(s)) \le d - \phi(s)\right\}.$$

This amounts to the so-called Robust Extended Kalman Filter (REKF) generalization presented in [9].

Acknowledgments

The authors would like to acknowledge Dang Xuan Thanh and Budi Mulyawan for the hardware implementation, experimental data collection, and their insights to this research. More information about the hardware implementation of the localization system can be found at http://www.cse.unsw. edu.au/~sensar/research/projects.

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom '99),* pp. 263-270, Aug. 1999.

[2] J.M. Kahn, R.H. Katz, and K.S. J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom '99)*, pp. 271-278, Aug. 1999.

[3] C.T. Incorporated, http://www.xbow.com.

[4] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," Technical Report IRB-TR-03-003, Intel Research Berkeley, Feb. 2003.

[5] G. Sibley, M.H. Rahimi, and G.S. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Ad-Hoc Sensor Networks," *Proc. Int'l Conf. Robotics and Automation (ICRA '02)*, vol. 2, pp. 1143-1148, May 2002.

[6] D.T.N.R. Group, *The Sammi Network,* http://www.cdt.luth.se/babylon/snc/.

[7] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with Zebranet," *Proc. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS-X '02),* pp. 96-107, Oct. 2002.

[8] R. Shah, S. Roy, S. Jain, and W. Burnette, "Datamules: Modeling a Three-Tier Architecture for Sparse Sensor Networks," *J. Elsevier Ad Hoc Networks,* vol. 1, nos. 2-3, pp. 215-233, 2003.

[9] I. Petersen and A. Savkin, *Robust Kalman Filtering for Signals and Systems with Large Uncertainties.* Boston: Birkhäuser, 1999.

[10] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," *IEEE J. Selected Areas in Comm.,* vol. 16, no. 6, pp. 922-936, Aug. 1998.

[11] M. Hellebrandt and R. Mathar, "Location Tracking of Mobiles in Cellular Radio Networks," *IEEE Trans. Vehicular Technology,* vol. 48, no. 5, pp. 1558-1562, Sept. 1999.

[12] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," *IEEE Personal Comm. Magazine,* vol. 7, no. 5, pp. 28-34, Oct. 2000.

[13] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location Support System," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom '00),* pp. 32-43, Aug. 2000.

[14] A. Savvides, C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MobiCom '01),* pp. 166-179, July 2001.

[15] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer,* vol. 34, no. 8, pp. 57-66, Aug. 2001.

[16] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, A. Morse, B. Anderson, and P. Belheumer, "Rigidity, Computation, and Randomization of Network Localization," *Proc. IEEE Conf. Computer Comm. (Infocom '04),* Mar. 2004.

[17] P. Bahl and V.N. Padmanabhan, "Radar: An In-Building User Location and Tracking System," *Proc. IEEE Conf. Computer Comm. (Infocom '00),* vol. 2, pp. 775-784, Mar. 2000.

[18] A. Savkin and I. Petersen, "Recursive State Estimation for Uncertain Systems with an Integral Quadratic Constraint," *IEEE Trans. Automatic Control,* vol. 40, no. 6, pp. 1080-1083, 1995.

[19] P. Pathirana and A. Savkin, "Precision Missile Guidance with Angle Only Measurements," *Proc. Information Decision and Control Conf.,* Feb. 2002.

[20] A. Savkin, P. Pathirana, and F. Faruqi, "The Problem of Precision Missile Guidance: LQR and H^{∞} Control Frameworks," *IEEE Trans. Aerospace and Electronic Systems,* vol. 39, no. 3, pp. 901-910, July 2003.

[21] H. Xia, "An Analytical Model for Predicting Path Loss in Urban And Suburban Environments," *Proc. Personal Indoor Radio Comm. (PIRMC '96),* 1996.

[22] K. Joanna, R. Heinzelman, and H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," *Wireless Networks,* 1999.

[23] T.M. Mote, http://www.xbow.com/Products/productsdetails.aspx?sid=61.

[24] Stargate, http://www.xbow.com/Products/productsdetails.aspx?sid=85.

[25] I. Lego Mindstorms, http://www.xbow.com.

[26] M. James and I. Petersen, "Nonlinear State Estimation for Uncertain Systems with an Integral Constraint," *IEEE Trans. Signal Processing,* vol. 46, no. 11, pp. 2926-2937, Nov. 1998.

[27] A. Savkin and R. Evans, *Hybrid Dynamical Systems: Controller and Sensor Switching Problems.* Boston: Birkhäuser, 2002.

Pubudu N. Pathirana received the BE degree with first class honors in electrical engineering, the BS degree in mathematics, and the PhD degree in electrical engineering from the University of Western Australia. In 1997/98, he worked as a research engineer in the industry in Singapore and in Sri Lanka. He was a postdoctoral research fellow at Oxford University in 2001 on a Ford Motor Company project, a research fellow in the school of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, and a consultant to the Defense Science and Technology organization (DSTO) Australia, 2002. He is currently with the School of Engineering and Technology, Deakin University, Australia. His research interests include missile guidance, autonomous systems, target tracking, computer integrated manufacturing, vision-based navigation systems, Quality of Service (QoS) management, and mobile/wireless internet.

Nirupama Bulusu received the PhD degree in computer science from the University of California, Los Angeles in 2002, the MS degree in computer science from the University of Southern California in 2000, and the BTech degree in computer science from the Indian Institute of Technology, Madras, India, in 1997. She is currently an assistant professor of computer science at the OGI School of Science and Engineering, Oregon Health and Sciences University (OHSU). She is a recipient of the Institute Merit Prize from IIT Madras, cofounder and program cochair for the IEEE EmNetS-I workshop, and the primary editor of the forthcoming book, *Wireless Sensor Networks: A Systems Perspective* (Artech House). Her research interests lie in networks and pervasive computing with an emphasis on wireless sensor networks and their applications. She is a member of the IEEE, the IEEE Computer Society, the ACM, and Sigma Xi.

Andrey V. Savkin received the MS degree in mathematics (1987) and the PhD degree in applied mathematics (1991) from the Leningrad State University, USSR. He is currently a professor with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, and the director for the Centre of Excellence in Guidance and Control. He has published four books and numerous journal and conference papers and served as an associate editor for several international journals and conferences. His current research interests include robust control and filtering, hybrid dynamical systems, missile guidance, networked control systems and control of networks, computer-integrated manufacturing, control of mobile robots, and application of control and signal processing to biomedical engineering and medicine. He is a senior member of the IEEE.

Sanjay Jha received the PhD degree from the University of Technology, Sydney, Australia. He is currently an associate professor at the School of Computer Science and Engineering, University of New South Wales, the head of the Network Research Group at CSE, and a project leader for the Smart Internet CRC. He was a visiting scholar at both the Distributed Computing and Communications Laboratory, Computer Science Department, Columbia University, New York, and the Transmission Systems Development Department of Fujitsu Australia Ltd, Sydney. He is the principal author of the book *Engineering Internet QoS* and coeditor of the book *Wireless Sensor Networks: A Systems Perspective.* His research interests include wireless ad hoc and sensor networking, quality of service (QoS), and active/programmable networks. He is a member of the IEEE.