Sequential Monte Carlo for Simultaneous Passive Device-Free Tracking and Sensor Localization Using Received Signal Strength Measurements

Xi Chen
McGill University
Montréal, Canada
xi.chen8@mail.mcgill.ca

Andrea Edelstein
McGill University
Montréal, Canada
andrea.edelstein@
mail.mcgill.ca

Yunpeng Li
Beijing Univ. Posts & Telecom.
Beijing, China
liyp@bupt.edu.cn

Mark Coates
McGill University
Montréal, Canada
mark.coates@mcgill.ca

Michael Rabbat
McGill University
Montréal, Canada
michael.rabbat@mcgill.ca

Aidong Men
Beijing Univ. Posts & Telecom.
Beijing, China
menad@bupt.edu.cn

ABSTRACT

This paper presents and evaluates a method for simultaneously tracking a target while localizing the sensor nodes of a passive device-free tracking system. The system uses received signal strength (RSS) measurements taken on the links connecting many nodes in a wireless sensor network, with nodes deployed such that the links overlap across the region. A target moving through the region attenuates links intersecting or nearby its path. At the same time, RSS measurements provide information about the relative locations of sensor nodes. We utilize the Sequential Monte Carlo (particle filtering) framework for tracking, and we use an online EM algorithm to simultaneously estimate static parameters (including the sensor locations, as well as model parameters including noise variance and attenuation strength of the target). Simultaneous tracking, online calibration and parameter estimation enable rapid deployment of a RSS-based device free localization system, e.g., in emergency response scenarios. Simulation results and experiments with a wireless sensor network testbed illustrate that the proposed tracking method performs well in a variety of settings.

Categories and Subject Descriptors

C.3 [**SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS**]: Signal processing systems; G.3 [**PROBABILITY AND STATISTICS**]: Probabilistic algorithms (including Monte Carlo)

General Terms

Algorithms, Measurement

Keywords

Target Tracking, Node Localization, Particle Filter, Device-Free, on-line EM, RSS

1. INTRODUCTION

In a wireless sensor network, signal attenuation occurs when a target moves between and around nodes in the network. For radio-frequency (RF) links connecting different pairs of nodes, the amount of attenuation seen on these links varies based on the proximity of the target to the links. These time-varying patterns of link attenuation provide information about the target location, allowing the network to track the target's motion. This procedure is referred to as RF tomography.

RF tomography is a promising technique with many practical applications. For example, it is often difficult for first responders to locate survivors in disaster situations such as fires or earthquakes [21]. RF tomography could be used to locate these survivors quickly without the need for responders to enter any structures, saving time and, potentially, lives. Similarly, RF tomography can be used for security and surveillance applications such as through-the-wall imaging and perimeter monitoring. RF tomography could be used in a smart home to control lighting, heating, and air-conditioning, saving power by sensing when there are no people in a room. It can also be used by doctors to track the movements of elderly patients remotely, either at home or at a hospital, without violating their privacy.

RF tomography is a passive, device-free [22] system. In particular, targets are not required to cooperate by transmitting a beacon or carrying, for instance, an RFID tag. The system is capable of detecting and tracking moving objects within the region of interest (the convex hull of the sensor nodes). In comparison to video-based (optical or infrared) surveillance systems, RF tomography does not require that the region of interest be illuminated [20]. Using wireless sensors also reduces the cost of the system in practical deployments, especially in small-scale target tracking scenarios, since it only requires several low-cost sensors and a laptop to process the data. Furthermore, it only takes a few minutes to place sensors around the area that needs

to be sensed. The ability to deploy a network easily and quickly makes RF tomography useful in the emergency response scenarios mentioned above.

In this paper, we propose and evaluate a novel algorithm for simultaneous RF tomographic tracking and sensor localization. Unlike existing RF tomography schemes, we do not assume that sensor locations are known a priori. We adopt a particle filtering (sequential Monte Carlo) approach. In order to make the algorithm computationally efficient and improve tracking accuracy, we introduce a new measurement model, whose form is validated by experimental data. This model does not quantize the region of interest into pixels, in contrast to the measurement models of [20, 21]. Our algorithm incorporates an on-line Expectation Maximization (EM) procedure to estimate the parameters of the dynamic model of the target and the observation model. These parameters can vary significantly for different targets and environments, so the on-line EM algorithm provides an important self-calibration mechanism. Received signal strength (RSS) measurements also encode information about the locations of the sensor nodes themselves. Leveraging this fact, we also incorporate a technique for simultaneously localizing sensors while tracking, where our localization algorithm corrects for the distortion to RSS-based distance estimates caused by the attenuating object whose motion we are tracking through the field. We validate our approach via both extensive simulations and experiments conducted with a wireless sensor network testbed.

1.1 Related Work

1.1.1 RF Tomographic Tracking

Many localization and tracking techniques have been proposed in the past, and they can be divided into two fundamental categories, device-based techniques and device-free techniques, depending on whether the targets carry tags or not. Device-based techniques such as GPS [22] and some RF-based systems [3] require the targets to cooperate and carry devices which can transmit beacon signals. This approach is not applicable in many of the scenarios mentioned above (e.g., emergency response situations like earthquakes or fires).

The term "device-free passive localization and tracking" (DFPLT) was coined by Youssef, Mah, and Agrawala [22] to describe the tracking systems that do not require the target to carry any device. DFPLT systems, such as RF tomography, use variations in received RF signals to detect and track targets moving through the region of interest. Many different DFPLT algorithms have been proposed in the past five years [12, 18, 22]. In contrast to the approach taken in this paper, these approaches require a significant initial training phase, with data gathered when there are targets at known locations. This makes rapid deployment, e.g., in emergency response scenarios, more challenging.

A number of other algorithms have been developed based on models of received signal strength [9,10,20,21,23]. These approaches eliminate the need for extensive training. By analyzing the changes in the RSS values, these algorithms can track the targets as soon as nodes are deployed in a new environment. Zhang et al. [23] use Mica2 sensors placed on the ceiling to localize the targets moving below. The Radio Tomographic Imaging (RTI) system proposed by Wilson and Patwari [21] makes use of image reconstruction methods to estimate a map of attenuation in the region of interest at sequential points in time. This work was extended in [20] to use the empirical RSS variance on each link, rather than the mean RSS, to determine the presence of the targets, and this promising approach has demonstrated the ability to track people through walls. The recent thesis [19] also describes a particle filter for RF tomographic tracking. That approach is based solely on sequential importance resampling, and model parameters such as noise variance, attenuator parameters, and sensor locations are assumed to be known. Our previous work [10] proposed a particle filtering approach for RF tomographic tracking under the assumption of known sensor locations. This paper builds on those previous results, incorporating a scheme to jointly estimate unknown sensor locations while tracking and providing a more extensive experimental evaluation.

1.1.2 Node Localization

Node localization is a fundamental problem in wireless sensor networks, and it has been studied extensively (see, e.g., [13] and references therein). RSS-based localization schemes model RSS as a decaying function of the distance between the transmitter and receiver, where the rate of decay (the path-loss parameter) is either assumed to be known or it is estimated from training data in the particular environment of interest. Several previously-proposed localization algorithms (such as those in [14] and [5]) rely on first estimating inter-node distances from noisy RSS measurements and then finding an embedding of the nodes in the plane that respects the estimated distances. This embedding can be made unique through the use of a few anchor nodes whose position is known *a priori* (e.g., via GPS). However, most previous work generally assumes a homogeneous environment (i.e., constant path-loss). Our previous work [8] proposes a method for node localization that directly accounts for the attenuating and scattering effects of objects which may be located between and around the nodes whose locations are being determined, assuming the locations of these attenuating objects have been estimated. The method proposed in this paper builds upon this previous work, incorporating the localization technique with the sequential Monte Carlo framework for tracking attenuating objects.

1.1.3 Simultaneous Localization and Mapping

The simultaneous localization and mapping (SLAM) problem has received significant attention in the robotics community (see [4, 7, 11]). In that problem, the robot must track its own state while also localizing landmarks and building a map, given observations of the world which are often translated to estimates of distances to the landmarks. Within the wireless sensor networks community, the problem of simultaneous localization and tracking (SLAT) has also been considered [1, 17]. There are high-level parallels between SLAM/SLAT and the problem considered in this paper, in that we are also simultaneously localizing based on measurements of the target relative to a number of reference points (sensor locations) which also need to be determined. However, there are also a number of differences which prevent one from directly applying SLAM/SLAT techniques in our problem setting. First, these methods typically treat the unknown parameters (landmark locations) as additional state variables within the filtering methodology. This leads to a number of known problems, since these locations are

static and thus have no meaningful dynamic model. Moreover, SLAM/SLAT techniques are based on the probabilistic modeling assumption that the landmark locations (corresponding to sensor locations in our problem) are conditionally independent given the target location. However, this is not the case in our setup, since each measurement corresponds to a link in the network, carrying information about the distance between the two sensors at the ends of the link and not the distance between the target and each sensor. Thus, the conditional independence modeling assumption does not apply.

1.2 Contributions

This paper presents and evaluates a novel RF tomography tracking algorithm integrated with node localization and model parameter estimation techniques. The proposed approach is based on a continuous model for the effects of attenuating objects on a given link which avoids quantizing the region of interest into pixels. The extent to which an attenuator affects a particular link is exponential in the attenuator's proximity to the link. This model is validated via measurements with a TelosB wireless testbed. We adopt a particle filtering (sequential Monte Carlo) approach for tracking. Specifically, we use a version of the auxiliary particle filter which incorporates an online EM algorithm for estimating model parameters. In this manner, while we are tracking the target location, we simultaneously estimate model parameters such as the noise variance, parameters in the target attenuation model, as well as the sensor locations. These parameters can vary significantly for different targets and environments, so the on-line EM algorithm provides an important self-calibration mechanism. We envision this approach being applied in scenarios where the network operators rapidly deploy the network and provide rough estimates of the locations where sensors were deployed (e.g., circling the approximate sensor location on a GUI, which could be interpreted as a Gaussian ellipsoid), and we use this prior information to initialize the sensor locations. We evaluate the proposed approach via both simulation and via experiments conducted with a wireless sensor network testbed.

1.3 Paper Organization

The rest of the paper is organized as follows. Section 2 provides a formal problem statement. Section 3 describes the attenuation measurement model and Section 4 introduces the on-line SMC localization and tracking algorithm. Simulation and experimental results for the combined approach are presented in Sections 5 and 6, respectively, and Section 7 makes concluding remarks.

2. PROBLEM STATEMENT

Received signal strength (RSS) measurements on the many links connecting nodes in a sensor network reflect information about both (1) the pair-wise distances between sensor nodes and (2) objects moving through the sensed region. In particular, obstructions inside the area can absorb, scatter or reflect part of the signals. As the target moves, different links will be affected, revealing information about the location of the target within the region. Moreover, the precise nature of the RSS measurements depends on a number of model parameters which are generally not known a priori. Our goal is to use measurements of RSS on the links between many pairs of nodes and over multiple time steps to jointly track a target moving through the region of interest, localize sensor nodes, and estimate model parameters such as the noise variance.

We consider a wireless sensor network of N nodes and $M = \frac{N^2 - N}{2}$ bidirectional links. In each measurement interval, the N nodes successively broadcast packets and all neighboring nodes measure the RSS. The RSS value of bidirectional link i at time step k is denoted $\gamma_i(k)$. We take $\gamma_i(k)$ to be the average of the RSS values along both the forward $\gamma_i^F(k)$ and reverse $\gamma_i^R(k)$ links: $\gamma_i(k) = \frac{1}{2}\big(\gamma_i^F(k) + \gamma_i^R(k)\big)$. The precise measurement model for $\gamma_i(k)$ is described in Section 3 below. Nodes successively broadcast packets at relatively small time intervals (e.g., every 5 ms), gathering RSS measurements which we stack into the vector $\boldsymbol{\gamma}(k)$. Under the RSS model adopted in this paper, $\gamma_i(k)$ can be split into three main terms: $\gamma_i(k) = \bar{\gamma}_i + z_k^i + \zeta_k^i$, where $\bar{\gamma}_i$ is the average RSS on link i when no target is present, z_k^i is the attenuation on link i at time step k, and ζ_k^i is additive white Gaussian noise affecting the measured RSS on link i at time step k. We assume that there is a window where we can gather measurements on all links when no target is present in order to estimate $\bar{\gamma}_i$, allowing us to estimate z_k^i from later measurements by subtracting off $\bar{\gamma}_i$ from $\gamma_i(k)$. Stacking these quantities into vector form, we have the measured RSS at time step k is $\boldsymbol{\gamma}(k)$, and the attenuation caused by the target is reflected in the differences $\mathbf{z}_k$.

Our goal is to track a single moving target described by state $\mathbf{x}_k$, with motion specified by a Markovian dynamic model $f(\mathbf{x}_k|\mathbf{x}_{k-1})$. In order to do this, we strive to maintain a particle approximation of the marginal posterior $p(\mathbf{x_k}|\mathbf{z}_{1:k})$ and estimate the expected value of $\mathbf{x}_k$ under this distribution. Simultaneously, we seek to estimate the unknown sensor locations, $\mathbf{s}$, and measurement model parameters θ from the RSS measurements $\boldsymbol{\gamma}(k)$, assuming that the locations of a few anchor nodes are known in advance.

3. MEASUREMENT MODEL

The measurement model describes the relationship between the true state, the sensor locations and the measurement values. Wilson et al. proposed measurement models for RF tomography in [20, 21]. Since these were employed in an imaging framework, the models were pixelized, i.e. the area under surveillance was divided into fixed size pixels. Our goal is tracking, not imaging, so there is no need to introduce pixels, and indeed such an introduction is undesirable, because it necessarily leads to additional quantization error. We therefore develop a pixel-free model that is better suited to the sequential Monte Carlo method we adopt, significantly enhancing the computational efficiency and leading to improved tracking accuracy.

The form of the proposed pixel-free model is motivated by experimental data recorded in a sensor network deployed in multiple outside environments with relatively few obstructions (some trees and a statue). The experiments involved a human walking around a region surrounded by sensor nodes (see Section 6 for more details of the sensor deployment).

For the bidirectional link i between a node pair, consider an ellipse with foci at the transmitter c and receiver e. Define

$$\lambda_k^i \triangleq d_i^c(\mathbf{x_k}) + d_i^e(\mathbf{x_k}) - d_i \qquad (1)$$

where $d_i^c(\mathbf{x_k})$, $d_i^e(\mathbf{x_k})$ are the distances from the target's position to the transmitter and receiver, respectively. The

parameter λ_k^i is equal to the major diameter of an ellipse passing through $\mathbf{x}_k$ with foci at the transmitter c and receiver e of the ith link, minus the length of ith link, d_i. Note that this quantity captures information about the position of link i at time step k.
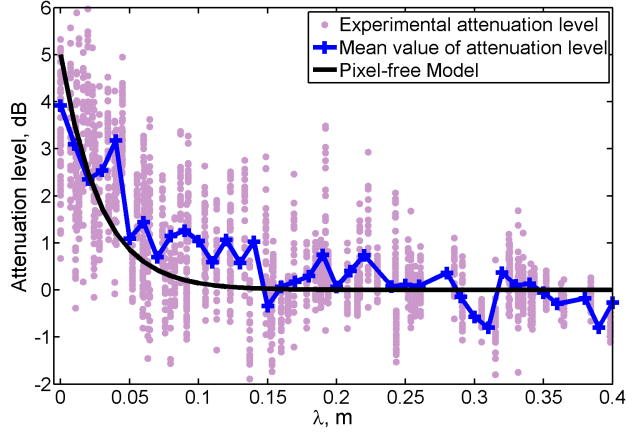

Figure 1: Attenuation level versus λ for the proposed pixel-free model (a comparison between the model and experimental measurements.)

For every measurement set, we calculated λ^i for all links and measured the corresponding attenuation z^i on the ith link (the measured RSS value minus the background mean, determined from a set of measurements conducted when the monitored area was empty). Figure 1 plots these attenuation values as a function of λ. Superimposed on the figure is the mean attenuation (calculated over all points within bins of λ-range 0.01) and our proposed model, detailed below. The model involves three parameters; in the graph these have been determined by using straightforward regression to minimize the mean-squared error.

Figure 1 suggests that the mean attenuation level decays approximately exponentially with λ. The individual attenuation levels are scattered around this mean and this can be reasonably captured by an additive Gaussian "noise". The pixel-free model for the attenuation caused by an object can then be described as follows:

$$\mathbf{z}_k = \phi \times \mathbf{g}_k + \sigma_s \mathbf{S}_k. \tag{2}$$

Here ϕ is the (modelled) value of mean attenuation at $\lambda = 0$ (i.e., when the target is in directly obstructing the link), and $\mathbf{S}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{M \times M})$ is additive white Gaussian noise; the parameter σ_s is the standard deviation that captures the variance of the noise, which is modelled as independent of λ. The $M \times 1$ vector $\mathbf{g}_k$ is defined as

$$\mathbf{g}_k \triangleq \left[\; g_k^1 \ldots g_k^i \ldots g_k^M \; \right]^T \tag{3}$$

where $g_k^i \triangleq \exp\{-\frac{\lambda_k^i}{2\sigma_\lambda}\}$. The parameter σ_λ controls the rate of decay of the mean attenuation with respect to λ.

The proposed attenuation measurement model has three unknown parameters, ϕ, σ_λ, and σ_s. After conducting a large number of experiments, we have concluded that the value of σ_λ that provides the best fit to the observed data varies little for different (human) targets and surveillance environments. We have observed considerably more variation

in the best-fit values of ϕ and σ_s.

In addition to the attenuation, the localization task requires a model for the raw received signal strength measurements. The log-distance path loss model [16] relates the distance d_i between two sensors to the measured RSS γ_i as

$$d_i = d_0 10^{(P_0 - \gamma_i)/(10 n_p)}. \tag{4}$$

Here P_0 is the received power at a reference distance d_0 and n_p is the path loss exponent. These three values are assumed known or measurable during a calibration phase.

In this model the effects of any attenuating objects on the RSS are ignored. We attempt to address this oversight by expanding (4) to

$$d_i = d_0 10^{(P_0 - \gamma_i - z_i - b_i)/(10 n_p)}, \tag{5}$$

where z_i represents the attenuation due to a moving target and b_i represents the attenuation due to background objects.

We can rearrange this equation to obtain the following model for the RSS measurement for link i at time k:

$$\boldsymbol{\gamma}_i(k) = P_0 - \mathbf{z}_k^i - \mathbf{b}_k^i - 10 n_p \log_{10}(d_i/d_0). \tag{6}$$

For the remainder of the paper, we assume we have access to a window of sensor measurements when there is no target moving through the sensed area. From these measurements we calculate an average background RSS vector γ_{avg} which contains the average RSS values on all M links. This vector captures the attenuation caused by stationary obstructions in the region of surveillance. During the tracking period, an instantaneous RSS vector γ_k is collected at time k, and we subtract the background RSS to obtain the vector $\mathbf{z}_k = \gamma_{\mathrm{avg}} - \gamma_k$ of RSS changes.

4. TRACKING AND LOCALIZATION

Our task is to localize the sensors and, at the same time, track the moving target. We adopt a Sequential Monte Carlo (particle filtering) framework to perform the tracking and use an online expectation-maximization (EM) approach to sequentially update the estimates of static parameters, which include the node locations, two of the parameters of the measurement model, ϕ and σ_s, and one parameter that represents the noise standard deviation of the motion model, σ_v. We fix the third parameter in the measurement model, σ_λ, to a constant value, which is estimated from experimental data.

We model the target dynamics using a one-tap autoregressive (AR-1) Gaussian model, i.e. $\mathbf{x}_{k+1} = a\mathbf{x}_k + \sigma_v \mathbf{v}_k$, where $\mathbf{x}_k$ is the target position in the 2D plane and $\mathbf{v}_k \sim \mathcal{N}(0, 1)$. The constant $a < 1$ models a (small) drift towards the center of the surveillance region; we choose a as a constant that is close to 1, so that the drift is very small. There are two main motivations for the adoption of this model: (i) it assumes little knowledge about the nature of the motion; (ii) the online EM methodology we adopt requires that the target process is stationary and ergodic (which eliminates a pure random walk process). We denote the static parameters $\theta = [\mathbf{s}, \phi, \sigma_s, \sigma_v]$, where $\mathbf{s}$ are the sensor locations.

4.1 Auxiliary Particle Filtering

We apply auxiliary particle filtering in the on-line SMC algorithm to track the marginal posterior distribution $p_\theta(\mathbf{x}_k|\mathbf{z}_{1:k})$. Here we provide a fairly brief description of the

auxiliary particle filter; please see [6] for more detail and an excellent discussion.

The auxiliary particle filter builds on the sequential importance resampling (SIR) particle filter, so we begin with a description of the SIR filter. It strives to calculate at each time step k a weighted particle approximation $\{\mathbf{x}_{1:k}^{(i)}, W_k^{(i)}\}_{i=1}^K$ to a posterior of interest $p(\mathbf{x}_{1:k}|\mathbf{z}_{1:k})$. To do this, it exploits the Markovian nature of the dynamic process and the conditional independence of the likelihood functions (observation $\mathbf{z}_k$ depends only on target state $\mathbf{x}_k$).

If the particle approximation is available at time $k-1$, then the particle filter can form an updated approximation for time k by (i) propagating (extending) the particles by sampling from an importance function $q(\mathbf{x}_k|\mathbf{x}_{k-1},\mathbf{z}_k)$; (ii) evaluating the likelihoods of the extended particles and updating the weights accordingly; and (iii) optionally resampling the particles to construct a particle set with more evenly distributed weights. The resampling procedure replicates particles with high weights and eliminates those with low weights.

The SIR filter can perform poorly if the importance function q does not adequately take into account the information available in the measurements $\mathbf{z}_k$. The auxiliary particle filter (APF), introduced in [15], modifies the sampling step in an attempt to improve performance. The filter calculates a *first-stage weight* $\rho_k^{(i)}$ for each particle based on how well the particle can explain the observations $\mathbf{z}_k$. Ideally, this weight should be a good approximation to the likelihood $p(\mathbf{z}_k|\mathbf{x}_{k-1}^{(i)}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})\,d\mathbf{x}_k$, i.e. $\rho_k^{(i)} = \widehat{p}(\mathbf{z}_k|\mathbf{x}_{k-1}^{(i)})$. The APF then resamples the particles $\mathbf{x}_{k-1}^{(i)}$ according to the first-stage weights. After the resampling step, the particles are propagated according to an importance function $q(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$ and the new weights are calculated. The APF optionally includes a second resampling step.

The algorithm is specified below in Algorithm 1. Although it is not an ideal choice because it can lead to unbounded variance in the estimates [6], we use the following first-stage weights: $\rho_k^{(i)} = p(\mathbf{z}_t|\mu_k^{(i)})$. Here $\mu_k^{(i)}$ is the mean of $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$; this was one of the suggested approaches in [15].

4.2 On-line EM

The auxiliary particle filter can only operate if the values θ are provided; since we do not have knowledge of these, we need to estimate them and it is desirable to do this online while tracking the target. We use an on-line EM algorithm to form estimates of the set of parameters $\theta = [\phi, \sigma_s, \sigma_v]$. We develop a procedure based on the generic method outlined in Section III.B of [2].

The on-line EM algorithm in [2] strives to maximize a pseudo-likelihood function in order to form point estimates of the parameters θ. Recursive maximization of the likelihood functions themselves, $p(z_{1:k}|\theta)$, would require estimation of statistics based on probability distributions whose dimension is growing in time. The substitution of the pseudo-likelihood leads to calculations in a fixed dimension.

The on-line EM algorithm updates the parameters every L time-steps. We define $\mathbf{X}_b \triangleq \mathbf{x}_{bL+1:(b+1)L}$ and $\mathbf{Z}_b \triangleq \mathbf{z}_{bL+1:(b+1)L}$, where b is the index of the block. The log pseudo-likelihood function employed in [2] is defined, for m

```
// Initialization at time k = 1
for i = 1, ..., K do
    Sample x₁⁽ⁱ⁾ ~ q₁(·);
    Set weights W̃₁⁽ⁱ⁾ = pθ(z₁|x₁⁽ⁱ⁾)p(x₁⁽ⁱ⁾) / q₁(x₁⁽ⁱ⁾);
end
Normalize weights W̃₁⁽ⁱ⁾ so that ∑ᵢ₌₁ᴷ W̃₁⁽ⁱ⁾ = 1;
// For times k > 1
for k = 2, ... do
    // First-stage weights
    for i = 1, ..., K do
        Calculate ρ_k⁽ⁱ⁾;
        Set W_k⁽ⁱ⁾ = W̃_{k-1}⁽ⁱ⁾ × ρ_k⁽ⁱ⁾;
    end
    // Resample
    Resample {x_{k-1}⁽ⁱ⁾, W_k⁽ⁱ⁾}_{i=1}^K to obtain {x'_{k-1}⁽ⁱ⁾, 1/K}_{i=1}^K;
    for i = 1, ..., K do
        Set x_{1:k-1}⁽ⁱ⁾ = x'_{1:k-1}⁽ⁱ⁾ and ρ_k⁽ⁱ⁾ = ρ'_k⁽ⁱ⁾;
        Sample x_k⁽ⁱ⁾ ~ q(x_k|x_{k-1}⁽ⁱ⁾);
        Set W̃_k⁽ⁱ⁾ = pθ(z_k|x_k⁽ⁱ⁾)p(x_k|x_{k-1}⁽ⁱ⁾) / (ρ_k⁽ⁱ⁾ q(x_k|x_{k-1}⁽ⁱ⁾));
    end
    Normalize weights W̃₁⁽ⁱ⁾ so that ∑ᵢ₌₁ᴷ W̃₁⁽ⁱ⁾ = 1;
    // Optional second resample
    Resample {x_{1:k}⁽ⁱ⁾, W̃_k⁽ⁱ⁾}_{i=1}^K to obtain {x_{1:k}⁽ⁱ⁾, 1/K}_{i=1}^K;
end
```

Algorithm 1: Auxiliary Particle Filter

blocks, as

$$l(\theta) = \sum_{b=1}^m \log p_\theta(\mathbf{Z}_b) \tag{7}$$

where

$$p_\theta(\mathbf{Z_b}) = \int_{\mathbf{X}^L} p_\theta(\mathbf{x}, \mathbf{Z}_b)\,d\mathbf{x}. \tag{8}$$

If the process $\mathbf{x}_k$ is stationary and ergodic, then it can be shown that the average log pseudo-likelihood satisfies

$$\bar{l}(\theta) = \int_{\mathbf{Z}^L} \log p_\theta(\mathbf{z}) p_{\theta^*}(\mathbf{z})\,d\mathbf{z} \tag{9}$$

where θ^* is the true value of θ. This implies that an algorithm that can maximize $\bar{l}(\theta)$ will identify the true value of θ. We therefore apply online EM to recursively maximize $\bar{l}(\theta)$ by updating the estimate of θ via

$$\theta_b = \arg\max_{\theta \in \Theta} Q(\theta, \theta_{b-1}) \tag{10}$$

where

$$Q(\theta, \theta_{b-1}) = \int_{\mathbf{X}^L \times \mathbf{Z}^L} \log(p_\theta(\mathbf{x}, \mathbf{z})) p_{\theta_{b-1}}(\mathbf{x}|\mathbf{z}) p_{\theta^*}(\mathbf{z})\,d\mathbf{x}\,d\mathbf{z}. \tag{11}$$

The direct computation of Q cannot be performed, but we can replace (10) by the update $\theta_b = \Lambda(\mathbf{\Omega}(\theta_{b-1}, \theta^*))$, where $\mathbf{\Omega}(\theta_b, \theta^*)$ is a set of sufficient statistics and Λ is a mapping function from the sufficient statistics $\mathbf{\Omega}(\theta, \theta^*)$ to the θ that maximizes Q.

Four sufficient statistics are required for the three parameters ϕ, σ_s, and σ_v. These are of the form

$$
\begin{aligned}
\mathbf{\Omega}(\theta_{b-1}, \theta^*) &= [\omega_1, \omega_2, \omega_3, \omega_4] \\
&= E_{\theta_{b-1}, \theta^*}[\psi_1, \psi_2, \psi_3, \psi_4].
\end{aligned}
\tag{12}
$$

The expectation is with respect to $p_{\theta_{b-1}}(\mathbf{x}|\mathbf{z})p_{\theta^*}(\mathbf{z})$ and

$$
\begin{aligned}
\psi_1(\mathbf{X}_b, \mathbf{Z}_b) &= \sum_{k=bL+2}^{(b+1)L} ((\mathbf{x_k} - \mathbf{x_{k-1}})^T(\mathbf{x_k} - \mathbf{x_{k-1}})) \\
\psi_2(\mathbf{X}_b, \mathbf{Z}_b) &= \sum_{k=bL+1}^{(b+1)L} ((\mathbf{z_k} - \phi\mathbf{g}_k)^T(\mathbf{z_k} - \phi g_k)) \\
\psi_3(\mathbf{X}_b, \mathbf{Z}_b) &= \sum_{k=bL+1}^{(b+1)L} (\mathbf{z_k^T} \mathbf{g}_k) \\
\psi_4(\mathbf{X}_b, \mathbf{Z}_b) &= \sum_{k=bL+1}^{(b+1)L} ||\mathbf{g}_k||_2^2.
\end{aligned}
$$

The maximization function Λ is defined as

$$
\sigma_{vb} = \sqrt{\frac{\omega_1(\theta_{b-1}, \theta^*)}{2(L-1)}}
\tag{13}
$$

$$
\sigma_{sb} = \sqrt{\frac{\omega_2(\theta_{b-1}, \theta^*)}{ML}}
\tag{14}
$$

$$
\phi_b = \frac{\omega_3(\theta_{b-1}, \theta^*)}{\omega_4(\theta_{b-1}, \theta^*)}.
\tag{15}
$$

The expectations cannot be computed, because they are with respect to a measure that involves the unknown true value θ^*. But the sufficient statistics can be recursively estimated. The ergodicity and stationarity assumptions for the process imply that the blocks $\mathbf{Z}_b$ are samples from $p_{\theta^*}(\mathbf{z})$ and they can therefore be used for Monte Carlo integration. We can thus form the following update of the statistics

$$
\hat{\mathbf{\Omega}}_b = (1 - \alpha_b)\hat{\mathbf{\Omega}}_{b-1} + \alpha_b E(\Phi(\mathbf{X}, \mathbf{Z}_b)|\mathbf{Z}_b)
\tag{16}
$$

where the expectation is with respect to $p_{\theta_{b-1}}(x|\mathbf{Z}_b)$. Setting $\alpha_b = 1/b$ ensures convergence of $\hat{\mathbf{\Omega}}_b$ to $\mathbf{\Omega}(\theta_b, \theta^*)$. The maximization step then becomes $\theta_b = \Lambda(\hat{\Omega}_b)$.

As one final approximation, since $E(\Phi(\mathbf{X}, \mathbf{Z}_b)|\mathbf{Z}_b)$ does not have an analytical solution, we can use importance sampling, using the particle tracks and weights calculated by the auxiliary particle filter

$$
\hat{\mathbf{\Omega}}_b = (1 - \alpha_b)\hat{\mathbf{\Omega}}_{b-1} + \alpha_b \sum_{m=1}^{K} W_b^{(m)} \psi(\mathbf{X_b^{(m)}}, \mathbf{Z_b}).
\tag{17}
$$

The estimation of the locations is addressed with a slightly different procedure because of the difficulty in identifying suitable sufficient statistics and maximization functions. For each block of measurements b, we minimize the difference between $\boldsymbol{\gamma}_b$, the matrix of measured RSS distances over the time window b, which is independent of $\mathbf{s}$, and $\mathbf{\Gamma}(\mathbf{s})$, the matrix of model-based RSS values according to

$$
\hat{\mathbf{s}}_\mathbf{b} = \underset{\mathbf{s}}{\arg\min} ||\mathbf{A} \circ (\mathbf{\Gamma}(\mathbf{s}) - \boldsymbol{\gamma})||_F^2
\tag{18}
$$

$$
= \underset{\mathbf{s}}{\arg\min} \left|\left| \mathbf{A} \circ \left(P_0 - 10n_p\log_{10}\frac{\mathbf{d}(\mathbf{s})}{d_0} - \phi_b\mathbf{g}(\hat{\mathbf{X}}_b, \mathbf{s}) - \boldsymbol{\gamma} \right) \right|\right|_F^2,
\tag{19}
$$

where $||.||_F$ denotes the Frobenius norm, $\circ$ denotes the Hadamard product, and $\mathbf{A}$ is a weighting matrix which assigns a weight to each link. The values $\phi_b\mathbf{g}(\hat{\mathbf{X}}_b, \mathbf{s})$ are the estimated attenuations due to the target over the block of time b, which are derived from the target position estimates generated by the particle filter and the estimated ϕ value. For each time k in this window, $\hat{\mathbf{x}}_k = \sum_{m=1}^{K} W_k^{(i)}\mathbf{x}_k^{(i)}$ and $\mathbf{X}_b = \{\hat{\mathbf{x}}_k\}$ for $k = (b-1)L, \ldots, bL$.

The weight matrix $\mathbf{A}$ can be used to bias the algorithm so that it assigns more confidence to the RSS measurements taken over certain links at the expense of others, if there is more certain knowledge about some sensor locations. In our algorithm, we solve (18) using a simple gradient descent procedure and we set the $\mathbf{A}$ matrix so that links which travel from a non-anchor node to an anchor node (or vice versa) are weighted more highly than links which travel between two non-anchor nodes.

4.3 The Combined Algorithm

The complete algorithm, combining the auxiliary particle filter and the on-line EM, is described in Algorithm 2 below. For the simulations and experimental results reported in this paper, we have used the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$ as the importance function q.

```
// Initialization
Sample θ₀ ~ q(θ) and set b = 1;
for k = 1, 2, ... do
    // Filtering
    {x_k^(i), W_k^(i)}_{i=1}^K = Auxiliary particle
    filter({x_{k-1}^(i), W_{k-1}^(i)}_{i=1}^K));

    // On-line EM and Sensor Localization
    if k mod L = 0 then
        // E-step
        for i = 1, ..., K do
            Calculate W_b^(i) = p_{θ_{b-1}}(X_b^(i)|Z_b) / q_{θ_{b-1}}(X_b^(i)|Z_b);
        end

        Normalize weights {W_b^(i)} such that Σ_{i=1}^K W_b^(i) = 1;
        Update
        Ω̂_b = (1 - α_b)Ω̂_{b-1} + α_b Σ_{m=1}^K W_b^(m)ψ(X_b^(m), Z_b);
        // M-step
        Set θ_b = Λ(Ω̂_b) and b = b + 1;
        // Sensor Localization
        Set ŝ_b =
        argmin_s ||A ∘ (P_0 - 10n_p log_10 d(s)/d_0 - φ_b g(X̂_b, s) - γ)||_F^2;
    end
end
```

Algorithm 2: SMC RF Tomographic Tracking

The complexity of SMC tracking algorithm is O(MN) per time step, where M is the number of links and N is the number of particles used for tracking. The on-line EM algorithm, which is only executed every L time-steps, has a complexity of O(LMN). In other words, the complexity of on-line EM algorithm is O(MN) during every execution. The comparable computational cost enables the tracking system to collect the data packets and to perform real-time processing using a standard off-the-shelf laptop (CPU: Core 2 Duo T5670 1.8GHz, RAM 1GB in our experiment).
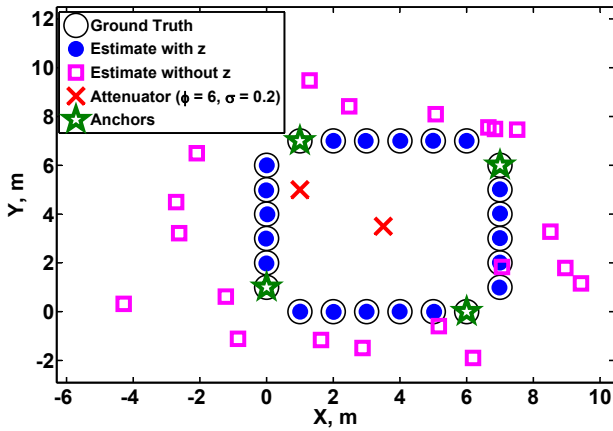
Figure 2: Simulated comparison of node localization, both taking z into account (for two attenuators, each with $\phi = 6$ and $\sigma_p = 0.2$) and ignoring it. The estimated locations for the nodes are pushed away from the ground truth.

5. SIMULATION RESULTS

In this section we present the results of simulations conducted to explore the performance of the proposed online SMC tracking and node localization methods. The simulation mimics a wireless sensor network with 24 sensor nodes. During the simulated period, a person walking within the network area follows a specific route at a speed of 0.5 m/s.

5.1 Node localization

We first present results for simulations exploring the performance of the proposed localization procedure, to highlight the importance of accounting for the presence of attenuators. When node locations are unknown, ignoring attenuators will bias the location estimates away from their true values. To illustrate this point, we compare the simulated performance of our localization algorithm with the widely-cited algorithm of Patwari et al. [14] for different values of σ_λ and ϕ in our model (see [8] for more details). In general, for small values of σ_λ and ϕ, our localization algorithm presents no significant advantage. This is only natural as these are cases where either the attenuator in the network does not have a very strong effect (small values of ϕ) or else where its effect is not very widespread (small values of σ_λ). However, as ϕ and σ_λ increase, our algorithm begins to offer significant improvements in localization. An example of one of our simulations can be seen in Figure 2. Here, we use a simulated network of 24 nodes (with the 4 corner nodes serving as anchors), placed evenly around a square of size 7 m × 7 m. Two attenuators, each with $\phi = 6$ and $\sigma_\lambda = 0.2$, are placed in the interior of the network. From this figure, we can easily see that if a localization algorithm does not account for the effects which obstructions have on RSS, its results will be skewed.

5.2 Joint Tracking and Localization

On-line SMC tracking with integrated node localization can be used when we only have location information about a small number of anchor nodes and the locations of the rest of the nodes remain unknown. We simulated a scenario

in which the motion of a single target was measured by a sensor network. The data was generated using the pixel-free model (with $\phi = 5$, $\sigma_s = 1$ and $\sigma_\lambda = 0.02$).

We present simulation results for two trajectories and two sensor layouts. The first sensor layout is a square of varying dimensions, with the four corner nodes being the anchor nodes with known locations. The second sensor layout is an irregular deployment. The four anchor nodes are approximately equally-spaced around the periphery. The trajectories include a square route (see Figure 3) and a zigzag route (see Figure 4). In each case, the target moves at a speed of 0.5 m/s and one complete set of measurements (all 24 sensors) is recorded every 120 milliseconds.
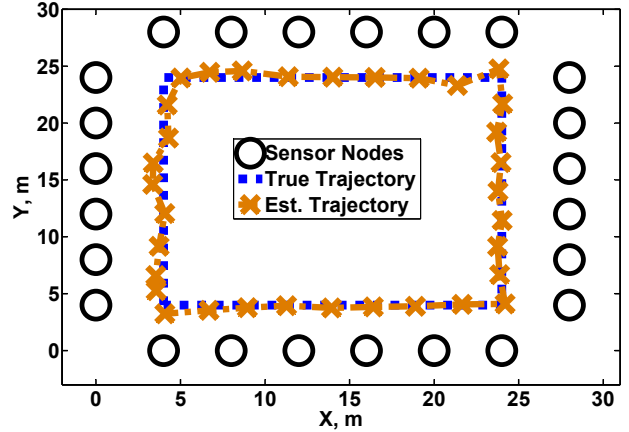

Figure 3: Tracking performance for a simulated example of simultaneous target tracking and localization. The target follows a square trajectory and sensors are arranged in a 28 m × 28 m square. The localization performance for this example is shown in Figure 5.
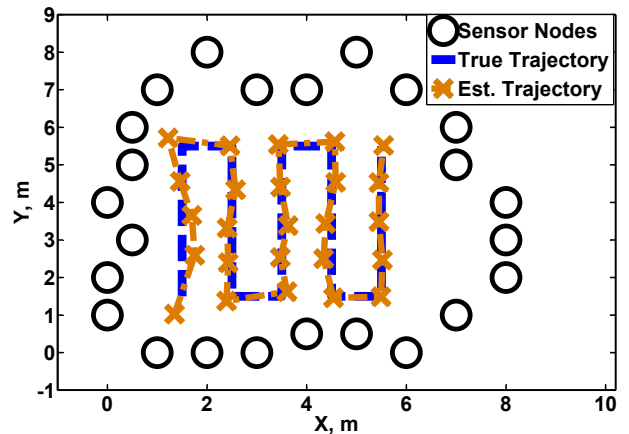

Figure 4: Tracking performance for a simulated example of simultaneous target tracking and localization. The target follows a zigzag trajectory and sensors are deployed in an irregularly-shaped "noisy square" (roughly 50 m^2).

For the non-anchor sensor nodes, we assume that the algorithm has some prior, imprecise information about the loca-
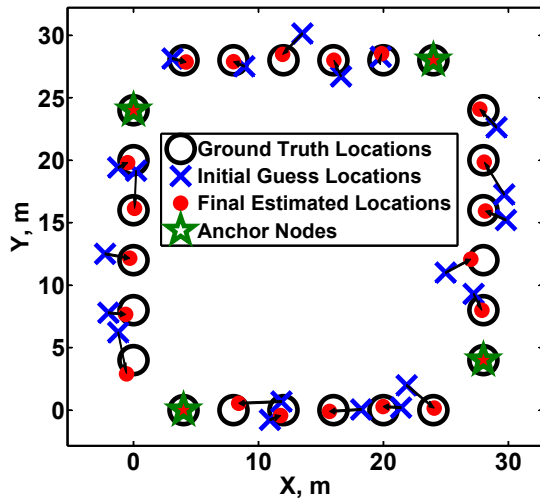
Figure 5: Localization performance for a simulated example of simultaneous target tracking and node localization. The target follows a square trajectory and sensors are deployed in a 28 m × 28 m square. The tracking performance for this example is shown in Figure 3.

tions. For example, if the sensors have been deployed around the perimeter of a building, we may have some knowledge about which of the nodes are on the left side of the building or the right side, even if we do not know precisely where they lie. In our simulations, we employ independent priors for the node locations, each being a two dimensional circularly-symmetric Gaussian with mean equal to the true location and standard deviation σ_p. In our simulations, we set $\sigma_p = 2$.

The tracking algorithm uses a Gaussian AR-1 dynamic model. The particle filter uses 1000 particles, and the unknown model parameters are initialized in the on-line EM algorithm by drawing from the uniform distributions: $\sigma_s \sim U(0, \sqrt{5}]$, $\phi \sim U(0, \sqrt{5}]$ and $\sigma_v \sim U(0, 1]$.

Figures 3 and 5 present examples of the tracking and localization performance for a 28 m × 28 m square layout with a square target trajectory. Figure 4 presents an example of tracking performance for the irregular layout (with an area of roughly 50 m^2) with a zigzag route. A complete set of sensor measurements is made once every 120 milliseconds; at this sampling rate, the depicted square route take 161 time steps to complete and the zigzag route takes 40 time steps.

In both cases, the on-line SMC tracking provides a good approximation of the ground-truth trajectory, even though we begin the algorithm with uncertainty about most of the node locations. The estimated trajectory follows the ground truth trajectory in straight lines, and experiences slightly higher error at the corners of the square route. Figure 5 shows the evolution in our knowledge of the node locations in the square route example. In this example, and in most realizations of the simulation, all of the final estimates are closer to the true positions than the initial guesses and the average location error is relatively low.

In Figure 6(a), we show the root MSE (RMSE) of the target tracking algorithm in the square layout for the square
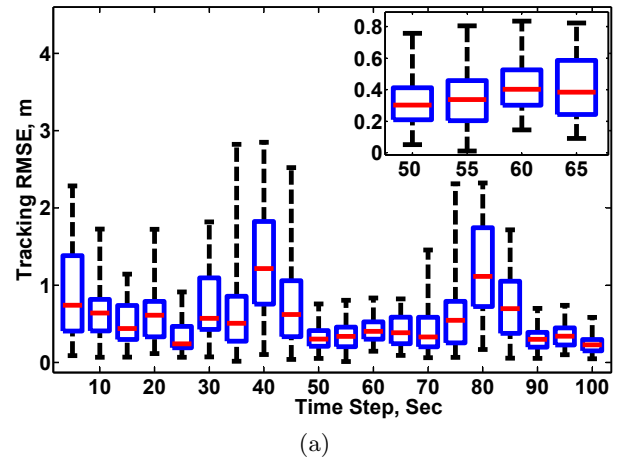
(a)
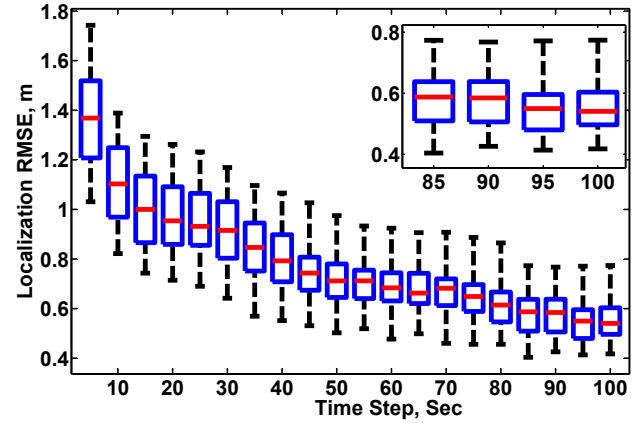
(b)

Figure 6: Box-and-whisker plot of RMSE as a function of time for a) tracking and b) node localization for a simulated target moving in a square trajectory within a 28 m × 28 m square using SMC tracking with initially-unknown node locations. The boxes range from the 25th to 75th quantiles, the whiskers extend 3 times the interquartile range, the median is marked as a line within the box, and the pluses indicate outliers.

target trajectory example, averaged over 100 realizations. This RMSE stays quite low (about 0.3 m) throughout most of the target's route in the 28 m × 28 m square. There are relatively higher RMSE values (on the order of 1 m) at the corners of the square trajectory. An abrupt turn is a much less likely event in the AR-1 model employed by the filter, so fewer particles are able to track the trajectory at the time step when the target changes direction at the corners. Only once out of 100 realizations did the algorithm lose track of the target.

Figure 6(b) shows the average RMSE/node of the node localization process in the square target trajectory example, again averaged over 100 realizations. Here, the RMSE/node starts out quite high and quickly decreases. The final localization RMSE is approximately 0.57 m (here we only show the first 100 time steps since the RMSE stays at roughly the same level thereafter).

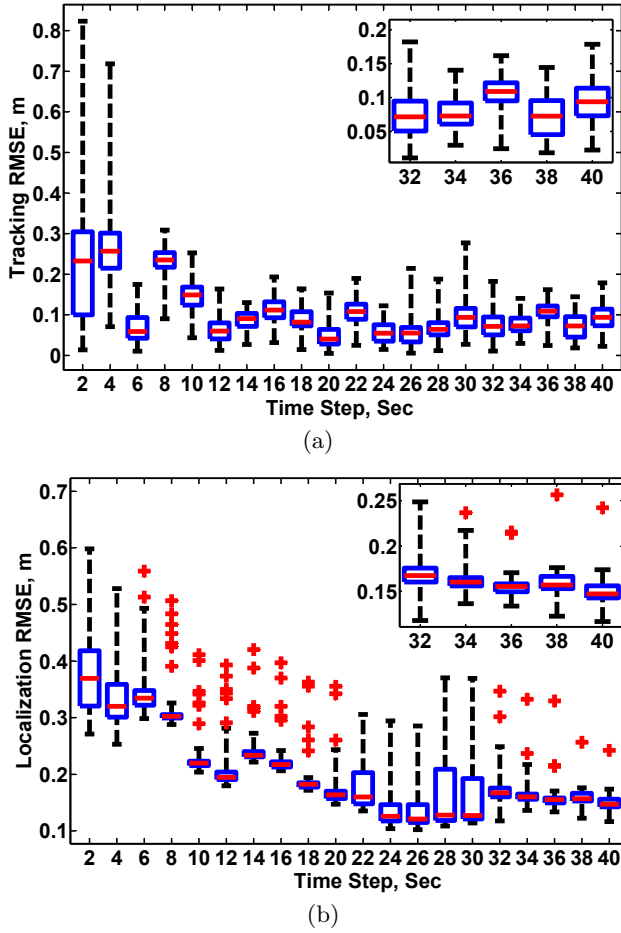Figure 7(b) depicts the RMSE results for the zigzag tra-

(a)

(b)

Figure 7: Box-and-whisker plot of RMSE as a function of time for a) tracking and b) node localization for a simulated target moving in a zigzag trajectory within an ≈ 50 m² irregularly-shaped area using SMC tracking with initially-unknown node locations. The boxes range from the 25th to 75th quantiles, the whiskers extend 3 times the interquartile range, the median is marked as a line within the box, and the pluses indicate outliers.

jectory. The median tracking RMSE is approximately 0.1 m after the first 10 time steps (once the online EM algorithm has approximately converged) and the final median node location error is approximately 0.17 m. The surveillance region is smaller than in the square layout, leading to a reduction in the error.

Using the aforementioned parameters for the measurement and dynamic models, we simulated different square layouts (ranging from 7 m × 7 m to 35 m × 35 m). The initial uncertainty in the node locations, governed by the value σ_p, was scaled in accordance with the area of the square. Tables 1 and 2 show, respectively, the RMSE of the target tracking and the average RMSE/node for the localization in these different scenarios. As expected, in both tables, the error increases slightly as the network size increases and, hence, as the density of the sensor nodes decreases. The accuracy is acceptable in all cases, however, with an average tracking RMSE of approximately 1m and a final localization

Network size	σ_p	1st step RMSE	Final step Average	RMSE RMSE
7 m × 7 m	1	0.5828	0.0860	0.2830
14 m × 14 m	$\sqrt{2}$	1.0107	0.3175	0.5722
21 m × 21 m	$\sqrt{3}$	1.3734	0.3621	0.6774
28 m × 28 m	2	1.9778	0.4538	0.8472
35 m × 35 m	$\sqrt{5}$	2.7020	0.3980	1.0028

Table 1: RMSE (in m) of SMC tracking between the first and final step. The RMSE over time represents an average of the RMSEs over all 161 time steps.

Network size	σ_p	First step RMSE	Final step RMSE
7 m × 7 m	1	0.7171	0.2520
14 m × 14 m	$\sqrt{2}$	1.3907	0.5519
21 m × 21 m	$\sqrt{3}$	1.9249	0.7398
28 m × 28 m	2	2.3500	0.8073
35 m × 35 m	$\sqrt{5}$	2.7166	0.7810

Table 2: Average RMSE/node (in m/node) of localization between the initial guess and the final estimation.

RMSE of 0.8 m in the case of a 35 m × 35 m network.

In Table 3, we see how the SMC tracking with integrated node-localization compares to tracking carried out with perfect a priori knowledge of the node locations. There is a clear performance penalty when the node locations are initially unknown. The RMSE values with unknown node locations are 6-7 times higher than those with known node locations. The estimation error experienced under unknown node locations scenarios is still practically acceptable (with a median error of 1 m for a relatively large network); note also that this is the average RMSE over all time steps. The average error towards the end of the trajectory, when node location and model parameter estimates are more accurate, is significantly less (median error of 0.4 m).

Network size	σ_p	RMSE with known node locations	RMSE with initially unknown node locations
7 m × 7 m	1	0.0436	0.2830
14 m × 14 m	$\sqrt{2}$	0.0728	0.5722
21 m × 21 m	$\sqrt{3}$	0.0975	0.6774
28 m × 28 m	2	0.1233	0.8472
35 m × 35 m	$\sqrt{5}$	0.1732	1.0028

Table 3: Comparison of the tracking RMSE (in m) averaged over all 161 time steps when node locations are known vs. when they are initially unknown.

6. EXPERIMENTAL RESULTS

This section presents an evaluation of the proposed algorithm using measurements from a wireless sensor network test bed. We conducted a measurement campaign, collecting RSS measurements with a set of 24 sensor nodes. All the nodes were Crossbow TelosB motes running TinyOS and using the IEEE 802.15.4 standard for communication in the 2.4 GHz frequency band. A simple token-ring transmission protocol was developed using nesC and each node was assigned
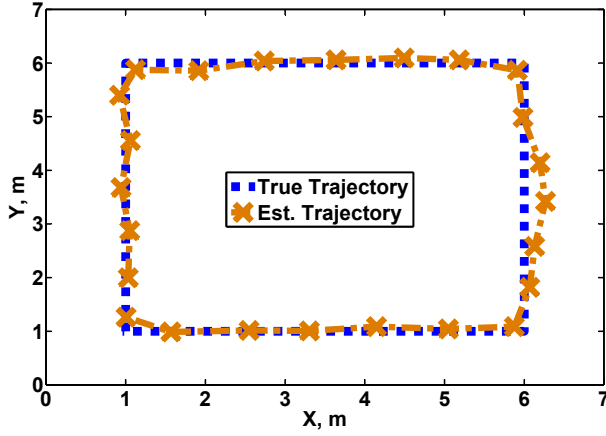
Figure 8: Experimental example of target tracking for square trajectory in a 7 m × 7 m square with a tree in the middle of it.

a fixed node ID at compile time. Data packets broadcast by each node contained this node ID along with the time of transmission and the measured inter-node RSS values which were received by that node from other nodes in the network. The interval between each transmission was set to 20 ms so that 50 samples were recorded every second.

The sensor network itself was constructed in a grassy outdoor field with a tree in the center of it (Field 1). The sensors were all placed on stands so that they were 1 m off the ground, and these stands were placed in a 7 m × 7 m square, mimicking the network we had heretofore been simulating. Markers were placed at 20 positions within the square so that the person walking through the network would be aware of—and be able to follow—the predetermined ground truth path. Before a person was brought into the network, however, the system sensed the vacant network area for roughly 3 minutes in order to generate the average RSS vector γ_{avg}. After this data had been collected, a person walked through the network following the ground truth path while the sensors collected more RSS measurements.

6.1 SMC Tracking with Known Node Locations

Figure 8 shows the results of tracking a person walking 25 times around a square (with known node locations), following a square trajectory in a 7 m × 7 m square in Field 1. In this particular experiment, the nodes were arranged such that there was a large tree in the center of the square. We compare the performance of the proposed particle filter to the Kalman filtering (KF) approach described in [20], where an image of attenuation is first estimated from the instantaneous link RSS measurements, and a Kalman filter is run over these estimated images. The RMSE of both the on-line SMC algorithm and imaging with KF are shown in Figure 9. The RMSE of both algorithms remains stable during the experiments. As shown in Figure 9(a), the tracking RMSE stays at about 0.3 m. Meanwhile, using the imaging with KF algorithm, the tracking RMSE in Figure 9(b) is about 0.6 m under the same condition.

We also carried out a similar tracking experiment for the same scenario in another field which had no tree in it (Field
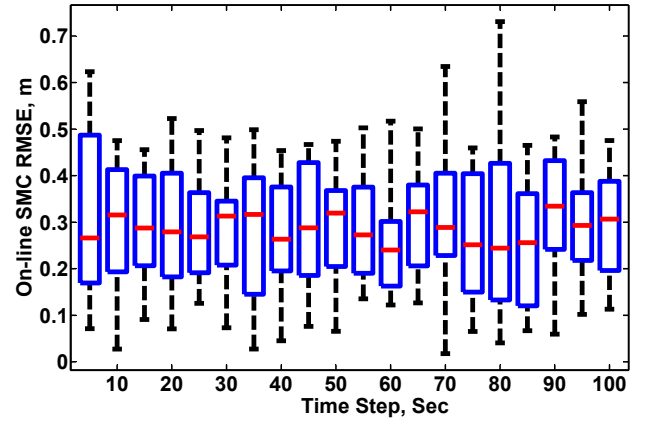
(a) Target tracking RMSE using the on-line SMC algorithm.
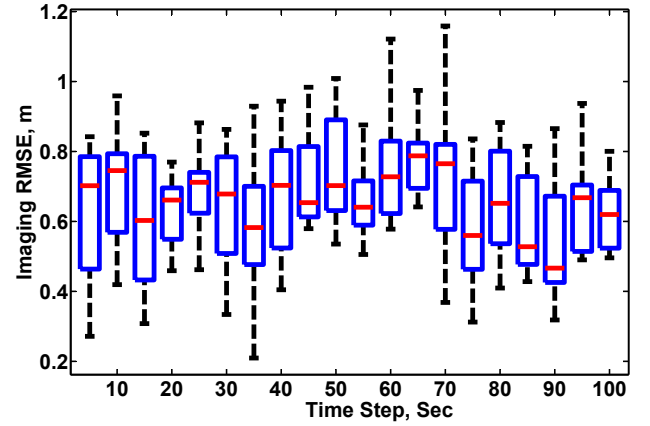
(b) Target tracking RMSE using the imaging with KF algorithm.

Figure 9: Box-and-whisker plot of RMSE as a function of time for the tracking of a real target moving in a square trajectory within a 7 m × 7 m square in Field 2 using both a) the on-line SMC algorithm and b) the imaging with KF algorithm with initially-unknown node locations. The boxes range from the 25th to 75th quantiles, the whiskers extend 3 times the interquartile range, the median is marked as a line within the box, and the pluses indicate outliers.

2). A numerical comparison of the tracking performance between Field 1 and Field 2 can be seen in Table 4. We observe that both algorithms perform better when there are no additional obstructions besides the one we are tracking and that on-line SMC outperforms imaging with KF in both scenarios. Here we only consider the performance of both algorithms based on experimental data; more detailed comparison of simulation results and interpretation can be found in [10].

In addition to the square trajectory scenario, we also present a scenario where the target moves along a zigzag route and where the sensors are arranged in a circular pattern with 7 m diameter, as shown in Figure 10. The average tracking RMSE is 0.2112 m when using on-line SMC and 0.4670 m when adopting the imaging with Kalman filtering method. Both Figure 8 and Figure 10 demonstrate that the proposed on-line SMC algorithm using the pixel-free model
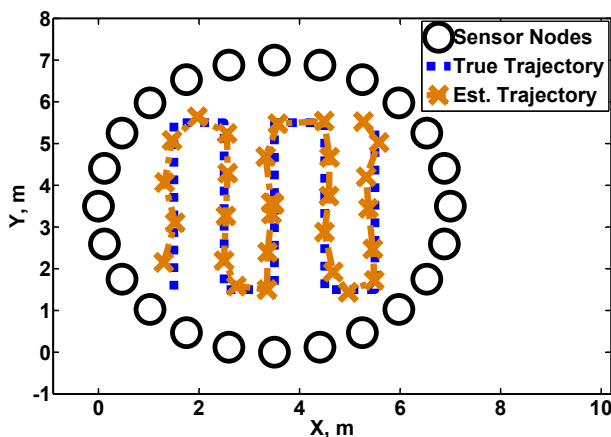
Figure 10: Experimental example of target tracking for a zigzag trajectory in a 7 m diameter circle.

Algorithm	RMSE for Field 1 (m)	RMSE for Field 2 (m)
SMC	0.4905	0.3214
Imaging with KF	0.8566	0.6404

Table 4: RMSE values for different algorithms run on the experimental results obtained in a 7 m × 7 m square area with a tree in the center.

achieves good trajectory estimates in different scenarios with experimental data, and it is capable of handling different sensor deployments.

7. CONCLUSIONS

This paper introduces a particle filtering method for RF tomographic tracking of a single target. The algorithm incorporates an on-line EM algorithm to estimate key parameters in the measurement and dynamic models. In order to improve the accuracy and to reduce computational requirements, we developed a pixel-free measurement model which is validated using experimental data. The proposed approach outperforms previously described approaches in the scenarios considered in this paper (outdoors, with few obstructions). Currently we are investigating robust methods for tracking and localization in more challenging indoor and through-wall scenarios, where multi-path and small-scale fading effects are more pronounced, as well as conducting a more detailed evaluation of the proposed joint localization and tracking methodology.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] N. Ahmed, M. Rutten, T. Bessell, S. Kanhere, N. Gordon, and S. Jha. Detection and tracking using particle-filter-based wireless sensor networks. *IEEE Trans. Mobile Computing*, 9(9):1332–1345, Sep. 2010.

[2] C. Andrieu, A. Doucet, and V. Tadic. On-line parameter estimation in general state-space models. In *Proc. IEEE Conf. on Decision and Control*, Seville, Spain, Jan. 2006.

[3] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. Int. Conf. IEEE Computer and Communications Societies (INFOCOM 2000)*, Tel-Aviv, Israel, Mar. 2000.

[4] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, 13(3):108–117, Sep. 2006.

[5] J. Costa, N. Patwari, and A. Hero. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):39–64, Feb. 2006.

[6] A. Doucet and M. Johansen. *Oxford Handbook of Nonlinear Filtering*, chapter A tutorial on particle filtering and smoothing: fifteen years later. Oxford University Press, 2010.

[7] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics and Automation Magazine*, 13(2):99–110, Jun. 2006.

[8] A. Edelstein, X. Chen, Y. Li, and M. Rabbat. RSS-Based Node Localization in the Presence of Attenuating Objects. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '11*, Prague, Czech Republic, May 2011. to appear.

[9] M. Kanso and M. Rabbat. Compressed RF Tomography for Wireless Sensor Networks: Centralized and Decentralized Approaches. In *Proc. IEEE Dist. Computing in Sensor Systems*, Santa Barbara, U.S.A., June 2010.

[10] Y. Li, X. Chen, M. Coates, and B. Yang. Sequential Monte Carlo Radio-Frequency Tomographic Tracking. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP '11*, Prague, Czech Republic, May 2011. to appear.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization adn mapping that provably converges. In *Intl. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, Aug. 2003.

[12] M. Moussa and M. Youssef. Smart devices for smart environments: Device-free passive detection in real environments. In *Proc. Int. Conf. Perv. Comp. and Comm.*, Galveston, TX, U.S.A., Mar. 2009.

[13] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal. Locating the nodes: Cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, Jul. 2005.

[14] N. Patwari, A. Hero, M. Perkins, N. Correal, and R. O'Dea. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal*

Processing, 51(8):2137–2148, Aug. 2003.

[15] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, Jun. 1999.

[16] T. Rappaport. *Wireless Communications Principles and Practice*. 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2002, pp. 161–166.

[17] C. Taylor, A. Rahimi, J. Bachrach, E. Shrobe, and A. Grue. Simultaneous localization, calibration, and tracking in an ad hoc sensor network. In *Proc. ACM/IEEE Conf. on Information Processing in Sensor Networks*, Nashville, TN, Apr. 2006.

[18] F. Viani, L. Lizzi, P. Rocca, M. Benedetti, M. Donelli, and A. Massa. Object tracking through RSSI measurements in wireless sensor networks. *Electronics Letters*, 44(10):653–654, May 2008.

[19] J. Wilson. *Device-Free Localization Using Received Signal Strength Measurements in Wireless Networks*. PhD thesis, University of Utah, Aug. 2010.

[20] J. Wilson and N. Patwari. See Through Walls: Motion Tracking Using Variance-Based Radio Tomography Networks. to appear, *IEEE Trans. Mobile Computing*, 2010.

[21] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *IEEE Trans. Mobile Computing*, 9(5):621–632, Jan. 2010.

[22] M. Youssef, M. Mah, and A. Agrawala. Challenges: device-free passive localization for wireless environments. In *Proc. Int. Conf. Mobile Computing and Networking*, Montreal, QC, Canada, Sept. 2007.

[23] D. Zhang, J. Ma, Q. Chen, and L. Ni. An RF-based system for tracking transceiver-free objects. In *Proc. IEEE Int. Conf. Perv. Comp. and Comm.*, White Plains, NY, U.S.A., Mar. 2007.