RSSI-Based Indoor Localization and Tracking Using Sigma-Point Kalman Smoothers

Anindya S. Paul, *Student Member, IEEE*, and Eric A. Wan, *Member, IEEE*

***Abstract*—Solutions for indoor tracking and localization have become more critical with recent advancement in context and location-aware technologies. The accuracy of explicit positioning sensors such as Global Positioning System (GPS) is often limited for indoor environments. In this paper, we evaluate the feasibility of building an indoor location tracking system that is cost effective for large scale deployments, can operate over existing Wi-Fi networks, and can provide flexibility to accommodate new sensor observations as they become available. This paper proposes a sigma-point Kalman smoother (SPKS)-based location and tracking algorithm as a superior alternative for indoor positioning. The proposed SPKS fuses a dynamic model of human walking with a number of low-cost sensor observations to track 2-D position and velocity. Available sensors include Wi-Fi received signal strength indication (RSSI), binary infra-red (IR) motion sensors, and binary foot-switches. Wi-Fi signal strength is measured using a receiver tag developed by Ekahau, Inc. The performance of the proposed algorithm is compared with a commercially available positioning engine, also developed by Ekahau, Inc. The superior accuracy of our approach over a number of trials is demonstrated.**

***Index Terms*—Bayesian inference, indoor tracking, received signal strength indication (RSSI)-based localization, sigma-point Kalman filter, sigma-point Kalman smoother, state estimation.**

I. Introduction and Related Work

LOCATION and context-aware technologies play a critical role in emerging next generation mobile applications. Example goals of these applications range from tracking assets within large warehouses, monitoring people inside assisted living communities, to adapting user interfaces based on location and activity. Key to each application is the ability to accurately localize and track an individual or asset. Explicit positioning sensors based on Global Positioning System (GPS) work worldwide and can sometimes achieve centimeter accuracy. However, GPS generally requires a direct view to several satellites, resulting in limited performance for indoor environments. Development of non-GPS-based solutions are thus of great interest for indoor use based on both existing signals and hardware, as well as new systems and sensor modalities. Additional design constraints pose significant challenges for development of such systems, including calibration overhead, user privacy, and the high variability of wireless channels.

A number of commercial systems and research prototypes currently exist for indoor localization. Systems typically use infrared (IR), ultrasound, or radio-frequency (RF) sensors [1]–[4]. Although they show potential for indoor tracking, each has its own limitations. The *Active Badge* System is one example of an early location-aware application [1]. The person to be tracked carries a small tag or "active badge," which emits a unique IR code every 15 seconds. A network of sensors pre-placed around the building pick up the periodic IR waveforms and a central "master-station" processes the data and triangulates the individual. Poor IR scalability and high maintenance overheads were some of the drawbacks faced by this system. The *Cricket* system places multiple "beacons" at several locations within the indoor environment which concurrently transmit RF and ultrasonic pulses. The person being tracked carries a listening device which uses time of flight (TOF) difference between RF and ultrasonic pulses in order to determine the distance to the beacons. Based on the TOF difference between multiple beacons, the closest beacon is inferred. Although Cricket improves accuracy and stability, high maintenance and calibration requirements require significant effort to use in practice. A commercially available system by *Sonitor*[5] also uses ultrasonic sensors for tracking a person. The user carries a small tag that emits its identification number via ultrasonics. Detectors scattered throughout the environment receives this information and triangulates the user. An advantage of this ultrasound based system is its immunity to interference and noise compared to RF, however the ultrasound cannot penetrate walls and is more expensive than comparable RF-based positioning systems.

RF based positioning systems are probably the most popular for indoor tracking. *RADAR* was one of the first RF signal strength-based positioning system used to track people inside buildings [2]. Multiple base stations (at least three) are placed with overlapping coverage within the area of interest. A laptop computer carried by an individual is used to collect the RF signal strength (RSSI) measurements. The system then compares the user's RSSI observation with a set of pre-stored signal strength measurements known as "fingerprints" at each of the base stations to identify the user's coordinates. The major disadvantages of the fingerprinting method include the need for dense training coverage and poor extrapolation to areas not covered during training. Although RADAR employs an empirical model for RF propagation and wall attenuation, actual RF signals deviate considerably while propagating indoors due to multipath,

Manuscript received July 01, 2008; revised February 10, 2009. Current version published October 21, 2009. This work was supported in part by Intel as part of the Behavioral and Intervention Commons (BAIC). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mohamed Sahmoudi.

The authors are with the Division of Biomedical Computer Science (BMCS), Department of Science and Engineering, School of Medicine (SOM), Oregon Health and Science University (OHSU), Beaverton, OR 97006 USA (e-mail: anindya@csee.ogi.edu; wane@ohsu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2009.2032309

metal reflection, and noise. Often indoor positioning systems have been designed to take advantage of public wireless local area networks (WLAN) instead of setting up proprietary RF networks. For example, *Place-Lab* uses publicly available 802.11 access points with receivers built into the users devices for positioning [6], [7]. The system compares the observed RSSI with a pre-stored *"radio map"* to determine the users position. Although it has the advantage of limited calibration requirements, reported accuracy is lower than existing positioning systems. Another example of WLAN based tracking includes the *"Horus WLAN location determination system"*[8]. A laptop computer carried by the user collects RSSI which is then compared to known RSSI fingerprints in order to perform localization. A commercially available product by *Ekahau* provides a complete tag and software solution using RSSI with the 802.11 protocol [9]. Multiple 802.11 access points are placed at predefined locations in the environment. The user carries a small tag that measures the RSSI at periodic intervals. The system compares the observed RSSI with a pre-stored set of RSSI collected during a separate training phase to compute the user's current location. The system is relatively inexpensive and energy efficient, but its accuracy is quite limited in certain scenarios as seen in the experimental section of this paper. Finally, *Ubisense*[10] is a tag based localization engine which uses ultra-wideband (UWB) radio technology to detect a mobile Ubisense tag. Multiple proprietary access points act as sensors that independently determine the *"angle-of-arrival"* of the UWB signal [11] and the *"time difference of arrival"* between a pair of sensors in order to perform positioning. While expensive and with significant calibration challenges, the Ubisense system can provide high tracking accuracy to within several centimeters.

Researchers have adopted a wide number of signal processing and pattern recognition-based algorithms to perform user localization. The fingerprinting based RSSI approach is one of the most widely used technology seen in the literature [2], [8], [12], [13]. Fingerprints are generated during an offline training phase where RSSI data is collected at a set of marked training locations. The most challenging aspect of the fingerprinting based method is to formulate a distance calculation that can measure similarity between the observed RSSI and the known RSSI fingerprints. Euclidean distance based calculation has been used in [14] to measure the minimum distance between the observed RSSI and the mean of the fingerprints collected at each training point. RADAR [2] uses a *k-nearest-neighbors* method in order to find the closest match between fingerprints and RSSI observation. Recently, research efforts have concentrated on developing a better distance measure that can take into account the variability of the RSSI training vectors. These methods estimate probability density for the training RSSI and then compute likelihood/*a posteriori* estimates during the tracking phase using the observed RSSI and the estimated densities [8], [15], [16]. User localization is performed using a maximum-likelihood (*ML*) or maximum *a posteriori* (*MAP*) estimate of position. Kernel based nonlinear distance calculations have also appeared in the literature for RSSI fingerprinting [12], [13]. Although these recent developments improve position estimates compared to simple *k-nearest-neighbors*, they often require substantially larger training sets and greater computational resources. In contrast to fingerprinting, model-based positioning techniques express the RF signal attenuation using a physics based "path loss" model [17], [18]. From an observed RSSI, these methods triangulate the person based on a distance calculation from multiple access points. However, the position-RSSI relationship is highly complex due to multipath, metal reflection, and interference noise [19]. Thus, the RSSI propagation may not be adequately captured by a fixed invariant model.

A number of variants on probabilistic Bayesian inference approaches have appeared in the literature [4], [7], [20]–[28]. Bayesian inference is a probabilistic framework which sequentially estimates the unknown state from noisy observations using a dynamic predictive model and an observation likelihood. Bayesian methods can estimate a person's velocity and acceleration in addition to position, and can also provide an uncertainty measure of the estimates. In [20], [25], the authors survey Bayesian filter implementations for location estimation using ultrasound, infrared, and laser range finders. They conclude that although particle filters can converge to the true posterior state distribution for non-Gaussian and multimodal cases, the Kalman filter and its variants are the most efficient in terms of memory and computation. Kalman filtering methods for real time positioning have long been popular in the robot tracking and navigation communities [29]–[31]. Recently, Kalman filter and their variants have also been applied to indoor people tracking. For examples, Fod *et al.* [22] and Hsieh *et al.* [23] describe a Kalman filter approach using multiple laser range finders. More recently, particle filters have been used to demonstrate encouraging performance, although at a high computational cost for real time people tracking [4], [26]–[28]. The particle filter-based system described by Hightower *et al.* [4] incorporates a random acceleration-based human motion model as the dynamics of the system, while the sensor model (observation likelihood) uses a single Gaussian with fixed predefined parameters. Letchner *et al.* [27] introduce a sensor measurement model in the particle filter framework that combines a Wi-Fi signal propagation model [32] and fingerprinting technique for localization. The method assumes radially symmetric attenuation of wireless signals and also requires large training data for fingerprinting. In addition, several algorithms assume an empirical *path-loss*-based radio signal propagation map to compute the likelihood of RSSI observation in a particle filter framework [6], [32], [33]. The performance of these algorithms, however, may degrade in practice due to RSSI variability over time and location. Recently, research efforts have been directed towards developing local RSSI likelihood models from the training data with known ground truth locations [26], [34]. Ferris *et al.* [26] use Gaussian processes to generate a measurement likelihood for wireless signal strength measurements in particle filter. However, learning the parameters using graph-based hyperparameter estimation can be slow and take significant computational resources. Also related to Bayesian inference are the use of hidden Markov modeling (HMM) approaches. The *Locadio system*[24] uses an HMM on a graph of location nodes to infer position based on the variation of the Wi-Fi signal strength. The person's motion is determined based on the variance of RSSI measurements

over a sliding window. However, significant RSSI variability (even at the same location) can cause a high number of errors in determining whether the person is moving or still.

Our approach to Bayesian inference is based on sigma-point Kalman filters (SPKF) [35]–[37]. SPKFs are estimation algorithms, which include the unscented Kalman filter (UKF) [38], central difference Kalman filter (CDKF) [39], and their square-root variants [40]. The SPKF has recently become a popular better alternative to the extended Kalman filter (EKF). The SPKF has been applied to a wide range of problems, including use in integrated navigation to combine GPS and inertial measurement units [41]–[43]. Additional background on the SPKF will be provided later in this paper. We use our recently proposed forward–backward statistically linearized sigma-point Kalman smoother (FBSL-SPKS) and fixed-lag sigma-point Kalman smoother (FL-SPKS) algorithms for tracking purposes [44]. Both the FBSL-SPKS and FL-SPKS estimators fuse a model of walking motion, room–wall configurations, and all available sensor observations in order to track a person. A random acceleration based model of human walking is used as the dynamic model of motion. This is augmented with a room–wall model involving a potential field created throughout the indoor environment in order to repel motion away from walls. Instead of using a fixed path loss based prior map for the observation model, we learn the position–RSSI relationship from the training data. Specifically, Radial-basis function (RBF) networks are used to provide a nonlinear mapping between known locations and observed RSSI values. These models are fit during a separate calibration process, and take into account the various multipath and other room specific characteristics. This article provides a more detailed description and analysis of our methods that was first presented in [45].

While our approach is generally independent of the specific hardware or sensor modality, the current system design uses RSSI sensors manufactured by Ekahau, Inc. The person(s) to be tracked carry a small body-borne device that periodically measures the RSSI at three or more standard Wi-Fi access points placed at predefined locations. Due to tag-based hardware limitations, the sampling rate is generally 4–8 s. The low sampling rate motivates the use of the sigma-point smoother over the filter implementation, as the smoother provides superior interpolation of data using both past and future observations. A smoother requires buffering of data and a fixed latency in performing the actual estimates. While the computational complexity is increased, the sigma-point smoothers can still be implemented far more efficiently than comparable particle smoother formulations. Augmenting the RSSI measurements are IR motion sensors mounted to the walls that provide a binary "on" signal when it detects a motion in its range. Similarly, binary foot-switches indicate the location of a person when stepped on. Experimentation were performed at several "living-laboratories" used to develop monitoring and assistive technologies for the elderly. The performance of our tracker was compared with the baseline Ekahau tracking engine. As will be shown, both the FBSL-SPKS and FL-SPKS based tracker provide significant improvement in position tracking accuracy.

The rest of the paper is organized as follows. Section II discusses the Bayesian framework and details the dynamic and observation models used in our SPKS framework. Section III examines the FBSL-SPKS and FL-SPKS-based tracking algorithms implemented in our system. Experimental results are given in Section IV, and finally discussion and conclusions in Section V.

II. Recursive Bayesian Estimation and Kalman Filtering

Recursive Bayesian estimation is a general probabilistic approach for sequentially estimating an unknown *state* probability density function over time using incoming noisy measurements and a mathematical process model. The problem can often be cast in terms of estimating the state of a discrete-time nonlinear dynamic system

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{1}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{n}_k) \tag{2}$$

where $\boldsymbol{x}_k$ represents the unobserved state of the system and $\boldsymbol{z}_k$ is the sensor observations. The process noise $\boldsymbol{v}_k$ drives the dynamic system, and the observation noise is given by $\boldsymbol{n}_k$. Note that we are not assuming additivity of the noise sources. The system dynamic model $\boldsymbol{f}(.)$ and observation model $\boldsymbol{h}(.)$ are assumed known.

A recursive solution to the Bayesian estimation problem for an arbitrary state space model is generally intractable. Sequential Monte Carlo (SMC)-based techniques, or particle filters, model the density of the state distribution using a set of discrete points and can provide arbitrary accuracy to the solution with a sufficient number of sample points. The optimal estimate of the state $\boldsymbol{x}_k$ in the MMSE sense is provided by the *"conditional mean"*, which may be calculated from the state posterior density updated recursively using new observations as they become available. Under a pure linear and Gaussian assumption, the Kalman filter provides an efficient and analytical solution for the posterior distribution [29], [37]. A first-order approximation to account for nonlinearities leads to the extended Kalman filter (EKF). While still assuming a Gaussian state distribution, the sigma-point Kalman filter (SPKF) provides superior performance over the EKF. Note that the SPKF refers to a family of related algorithms, the unscented Kalman filter (UKF), central difference Kalman filter (CDKF), and several variants. We will return to the details of the SPKF after first describing both the dynamic and observation models chosen for the specific indoor tracking application.

A. Dynamic Model

We define the state vector $\boldsymbol{x} = [x \quad y \quad v_x \quad v_y]^T$, corresponding to 2-D position and velocity for tracking purposes. A simple random walk model on the velocities [46], [47] is used for predicting walking motion. This is augmented with a room model involving a potential field created throughout the indoor environment in order to repel estimated motion away from walls.

The potential field can be created offline using prior knowledge of wall configurations and large furniture location. Computationally, this is achieved by dividing the space into

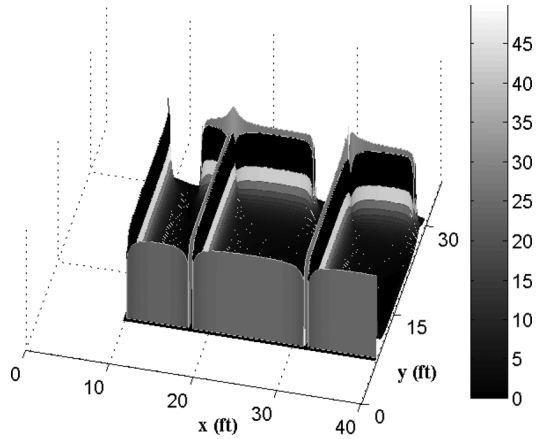


Fig. 1. Potential field is shown in a multi-room environment. As illustrated, the magnitude of the potential force peaks at the edge of the walls and decreases exponentially.

1-inch square cells. Each cell contains a binary certainty measure $C(i,j)$ that indicates whether the cell is occupied, i.e., an obstacle exits within the cell. The force $\boldsymbol{F}_{i,j}(x,y)$ exerted on a person due to an occupied cell is made inversely proportional to the distance between the person's current position and the occupied cell position [48]

$$\boldsymbol{F}_{i,j}(x,y) = -\frac{F_{\rm cr} C(i,j)}{d_{i,j}^2(x,y)} \left(\frac{x - x_c^i}{d_{i,j}(x,y)} \vec{\boldsymbol{x}} + \frac{y - y_c^j}{d_{i,j}(x,y)} \vec{\boldsymbol{y}} \right) \quad (3)$$

where $d_{i,j}(x,y)$ is the distance between the person's current position, (x,y), and the occupied cell position, (x_c^i, y_c^j). $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$ are the unit vectors along the x and y direction. $F_{\rm cr}$ is the force constant and design parameter that controls the overall strength of the repulsive force. If the force is too strong, location estimates will not be near walls or furniture. If the force is too small, tracking may result in trajectory estimates that pass through walls.

The total resultant force $\boldsymbol{F}_{\rm r}(x,y) = [\, F_x(x,y) \quad F_y(x,y) \,]$ is the vectorial sum of forces exerted by all the occupied cells on the person's current cell location

$$\boldsymbol{F}_{\rm r}(x,y) = \sum_{i,j} \boldsymbol{F}_{i,j}(x,y). \quad (4)$$

This repelling force function $\boldsymbol{F}_{\rm r}(x,y)$ is calculated once offline, and may be viewed as a potential field or simply a nonlinear function of the person's current position. Fig. 1 displays the corresponding magnitude of the potential field for a simple multi-room example.

Combining the potential field and a random walk model yields the dynamic state-space model [45]

$$x_{k+1} = x_k + \delta T v_{x_k} + \frac{\delta T^2}{2} F_{x_k}(x_k, y_k) \quad (5)$$

$$y_{k+1} = y_k + \delta T v_{y_k} + \frac{\delta T^2}{2} F_{y_k}(x_k, y_k) \quad (6)$$

$$v_{x_{k+1}} = \lambda v_{x_k} + \delta T F_{x_k}(x_k, y_k) + (1-\lambda)\, v_{p_{x,k}} \quad (7)$$

$$v_{y_{k+1}} = \lambda v_{y_k} + \delta T F_{y_k}(x_k, y_k) + (1-\lambda)\, v_{p_{y,k}}. \quad (8)$$

The parameter λ smooths the changes in velocities and also ensures that the variance of random process remains bounded. The integration time in this case is $\delta T = 1$ second. The process noise $\boldsymbol{v}_{p,k} = [\, v_{p_{x,k}} \quad v_{p_{y,k}} \,]$ is modeled as zero mean white Gaussian.

B. Observation Model

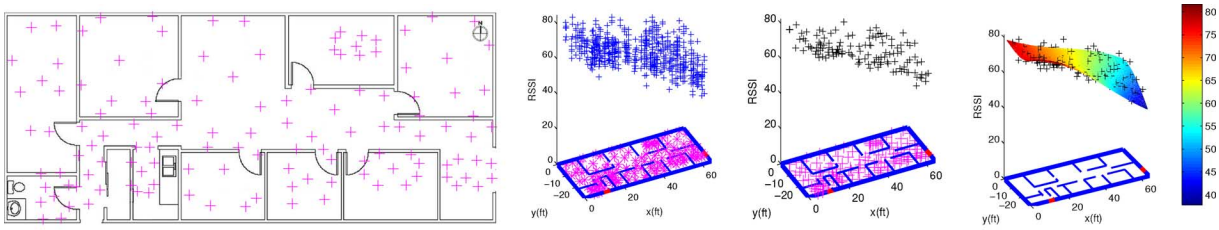
As we have available three different sensor technologies, the observation model in (2) depends on the specific technology used.

1) RSSI Observation Model: A naive approach to using RSSI measurements involves comparing an observed RSSI value to a table of previously recorded RSSI values and their associated positions. This direct "table look-up" approach, however, is prone to errors due to the high variability of RSSI values. In the Bayesian framework, the observation function can be viewed as a *generative* model providing the likelihood of a RSSI observation given the current estimate of the state position. In most RSSI tracking literature, the observation likelihood is approximated with a simple fixed *a priori* distribution (e.g., Gaussian distribution) [4], [7]. In our method, we characterize the RSSI-position relationship and variability by fitting nonlinear mappings between position and observed RSSI values.

Data to fit the maps are first collected during a calibration phase. This involves dividing the floor plan into P rooms or sections. In each section, the vertices and center of an approximate octagonal grid are used as calibration points. See Fig. 2(a) for illustration purposes. This calibration scheme was chosen to match the grid pattern used by the Ekahau positioning engine in order to allow for the direct comparisons of final performance. Note that an exact octagonal grid is not possible due to the presence of furniture, walls and other objects in the floorplan. At each calibration point, a person carrying a body borne RSSI tag spends $\boldsymbol{T}_c$ seconds (generally 60 seconds) while RSSI data is collected. Typically, RSSI values are recorded from M (generally 3–5) Wi-Fi access points located in the corners of the entire space to be calibrated. The person also performs a slow rotation at each point to average RSSI variability due to tag orientation. Note that if multiple tags are to be calibrated simultaneously, it is advisable to physically separate the tags on the person as far as possible, as we have found that multiple tags can interfere with RSSI consistency. This process is repeated at all calibration points in the space. Fig. 2(b) illustrates the collection of raw RSSI data at each calibration point. The number of RSSI samples collected at each calibration point are denoted as $\boldsymbol{N}_{\rm r}$ (generally 8-10). The $\boldsymbol{N}_{\rm r}$ number of RSSI samples are then averaged to obtain a representative mean RSSI observation per calibration point as shown in Fig. 2(c). Specific values for the amount of data collected, variability, etc., are tag specific and will be given in the experimental results section.

After RSSI data collection, an *RBF* network is used to fit a nonlinear map between known calibration locations and the mean RSSI observations as illustrated in Fig. 2(d). An *RBF* network is a feed forward neural network consisting of a hidden layer of radial kernels and an output layer of linear neurons [49]. A Gaussian kernel is used as the radial basis. This *RBF* map represents the forward generative observation model

$$z_{m,k} = h_{m,k}(x_k, y_k) + n_{m,k}^{\rm r} \quad (9)$$



(a) (b) (c) (d)

Fig. 2. (a) Example floor plan with calibration locations indicated by a "+", (b) Raw RSSI values recorded at each access point during calibration, (c) RSSI mean values at each calibration location, (d) RBF nonlinear map plotted with the RSSI mean values.

where $z_{m,k}$ is the observed RSSI from access point $m, 1 \leq m \leq M$, with noise $n^{\mathrm{r}}_{m,k}$ assumed to be Gaussian with zero mean and standard deviation equal to the RSSI variability determined from the calibration data. The RBF observation map $h_{m,k}$ for the access point m is specified by

$$h_{m,k}(x_k, y_k) = \boldsymbol{W}_m^T \boldsymbol{K}_{m,\mathrm{G}}([x_k \quad y_k]; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

where $\boldsymbol{K}_{m,\mathrm{G}}$ is the Gaussian kernel function [49] with mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$

$$\boldsymbol{\mu}_m = [\boldsymbol{\mu}_{m,1} \quad \boldsymbol{\mu}_{m,2} \quad \cdots \quad \boldsymbol{\mu}_{m,\mathrm{C}}]^T$$

$$\boldsymbol{\Sigma}_m = \begin{bmatrix} \boldsymbol{\Sigma}_{m,1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_{m,2} & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & \cdots & 0 & \boldsymbol{\Sigma}_{m,\mathrm{C}} \end{bmatrix}$$

where C is the number of Gaussian kernels in the hidden layer of the RBF network and $\boldsymbol{W}_m$ are the output layer linear weights

$$\boldsymbol{W}_m = [w_{m,0} \quad w_{m,1} \quad \cdots \quad w_{m,\mathrm{C}-1}].$$

The parameters of each Gaussian kernel $\{\boldsymbol{\mu}_{m,c}, \boldsymbol{\Sigma}_{m,c}\}$ and the hidden to output layer weights $w_{m,c}$ are learned using a hybrid procedure that operates in two stages. The prior weight, center position and the spread parameter of each Gaussian are first obtained by modeling the known calibration locations with a Gaussian mixture model (GMM) using the expectation–maximization (EM) algorithm. The output layer weights $\boldsymbol{W}_m$ are then calculated in a batch least-squares manner in order to minimize the MSE error at the output. Fig. 2(d) illustrates a nonlinear observation map learned from the calibration data.

The observed RSSI $z_{m,k}$, RBF function $h_{m,k}$, and the observation noise $n^{\mathrm{r}}_{m,k}$ from each access point are combined to form a multidimensional observation model

$$\boldsymbol{z}_k = [z_{1,k} \quad z_{2,k} \quad \cdots \quad z_{m,k} \quad \cdots \quad z_{\mathrm{M},k}] \tag{10}$$

$$\boldsymbol{h}_k = [h_{1,k} \quad h_{2,k} \quad \cdots \quad h_{m,k} \quad \cdots \quad h_{\mathrm{M},k}] \tag{11}$$

$$\boldsymbol{n}^{\mathrm{r}}_k = [n^{\mathrm{r}}_{1,k} \quad n^{\mathrm{r}}_{2,k} \quad \cdots \quad n^{\mathrm{r}}_{m,k} \quad \cdots \quad n^{\mathrm{r}}_{\mathrm{M},k}] \tag{12}$$

where $\boldsymbol{z}_k$ is the multidimensional RSSI observations emanating from each access point. Similarly $\boldsymbol{h}_k$ and $\boldsymbol{n}^{\mathrm{r}}_k$ are the augmented RBF observation model and the measurement noise for access points $1 \leq m \leq \mathrm{M}$.

Once fit using calibration data, this RBF observation model may be used in the Bayesian framework for tracking. As the RBF network is trained to learn a nonlinear mapping between known calibration locations and observed RSSI values, this model takes into account room specific multi-path and non-line-of-sight (NLOS) RSSI propagation. By learning the map, the need to specify the location of the access points is also avoided.

2) IR Motion Sensor Observation Model: Infrared (IR) motion sensors may be mounted to the walls and provide binary "on" signals when motion is detected within range. Localization using motion sensors are challenging due to their large beam width and high false alarm rate. The likelihood model for a motion sensor is modeled simply as a Gaussian distribution. The observation model is thus linear and defined as

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}^{\mathrm{ms}}_k \tag{13}$$

where $\boldsymbol{H}_k$ is the observation matrix

$$\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

and $\boldsymbol{n}^{\mathrm{ms}}_k$ is the Gaussian observation noise with mean and variance associated with the IR sensor. The mean value is taken to be a position inline with the orientation of the sensor at a distance based on the approximate sensor range. The variance is based on the beam width of the sensor. Specific values for the mean and variance are found by approximate characterization of the sensors. Filter performance is not highly sensitive to these values. Note that this simple model clearly does not take into account the specific geometry of the beam pattern, or other complicating factors such as memory and latency in the binary sensor. Incorporating a more accurate distribution would require a non-Gaussian framework (e.g., particle filters), and was not explored for this current phase of the research.

3) Binary Foot-Switch Observation Model: Similar to IR motion detectors, foot-switches may be placed on the floor to provide a binary "on" signal that indicates the location of a person. The observation likelihood may again be modeled simply as a Gaussian distribution with the corresponding observation model

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}^{\mathrm{f}}_k \tag{15}$$

where $\boldsymbol{H}_k$ is the observation matrix

$$\boldsymbol{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{16}$$

and $\boldsymbol{n}^{\mathrm{f}}_k$ is the Gaussian observation noise for the foot-switch sensors. The mean value of the Gaussian is set to the known loca-

tion of the switch. We set the variance σ_f^2 of the foot-switches to be approximately 10 ft in our experiments. While this is clearly larger than necessary, the aim was to simulate accuracy closer to that of the IR motion sensors during initial setup of the testing facilities. Setting the variance to a small value can provide exact localization at precise moments in time, but can also lead to "jumps" in trajectories at the vicinity of the foot-switches. Artificially increasing the variance helps ensure the smoothness of the estimated trajectory.

4) Multiple Sensors Observation Model: The Kalman framework allows for fusing multiple sensors of different types as available. An augmented observation vector is specified

$$\boldsymbol{z}_k = [\, z_k^{\mathrm{RSSI}} \quad z_k^{\mathrm{IR}} \quad z_k^{\mathrm{Foot}} \,] \tag{17}$$

along with the corresponding observation functions. Note that the dimension of this augmented observation may change at each time step to account for varying sensor sampling rates or missing observations.

III. SPKS-Based Location Tracker

The Kalman filter provides the optimal Bayesian recursive estimate in the MMSE sense for the state $\boldsymbol{x}_k$ of a linear state-space system driven by Gaussian noise [50]. The estimate is optimal given all noisy measurements $\boldsymbol{Z}_k = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k]$ up to the current time index k. For nonlinear state-space models, the extended Kalman filter (EKF) approach may be used whereby the nonlinear state-space model is linearized around the estimate of the current state using a first order Taylor series expansion. The sigma-point Kalman filter (SPKF), which includes the unscented Kalman filter (UKF) [51], central difference Kalman filter (CDKF) [39], and their square-root variants [40], has recently become a popular better alternative to the EKF. Like the EKF, the SPKF approximates the state distribution by a Gaussian random variable (GRV). However, the probability distribution is represented by a set of carefully chosen deterministic sample points (called sigma points). These sigma points are then propagated through the true nonlinear system, with the posterior mean and covariance calculated using simple weighted averaging. Although the order of computational complexity of both the SPKF and EKF is equal, this approach captures the posterior mean and covariance accurately to the second-order (third-order is achieved for symmetric distribution) compared to the EKF, which linearizes the nonlinear systems and only achieves first-order accuracy.

While the SPKF may be applied directly to the tracking problem, we have found improved performance through the use of the sigma-point Kalman smoother (SPKS). The Kalman smoother estimates the conditional expectation of the state at time k given all past and future measurements $\boldsymbol{Z}_k = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{\mathrm{N}}]$, $1 \leq k \leq N$. Several common Kalman smoothing formulations are given in [29] and [52]–[55]. In this article, two SPKS variants are investigated, the forward–backward SPKS (FBSL-SPKS) [44] and fixed-lag SPKS (FL-SPKS) [52]. The FBSL-SPKS corresponds to a fixed interval smoothing approach whereby the final time N is fixed and smoothed estimates are found using all N measurements. In FBSL-SPKS, a standard SPKF is used to estimate the states forward in time. A modified backward SPKF is then used to estimate the states in the backward direction, which are combined with the forward estimates to calculate the smoothed estimates. In the FL-SPKS, the estimate of the state at each time step is calculated after waiting for L future measurements. The SPKF is reformulated by incorporating a large state vector augmenting states from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k-L}$. Details of both these smoother variants are described in the following sections.

A. Forward–Backward Statistical Linearized Sigma-Point Kalman Smoother (FBSL-SPKS)

In the FBSL-SPKS, a standard SPKF is run in the forward direction using the nonlinear state-space model in (9). A backward filter then computes the estimates operating on the inverse dynamics of the forward filter. As the forward nonlinear dynamics are never analytically linearized, the backward filter is not well defined for a nonlinear dynamical system. To derive the appropriate backward filter, the SPKS makes use of the weighted statistical linear regression (WSLR) formulation of the filter. WSLR is a linearization technique that takes into account the uncertainty of the prior random variable (RV) when performing local linearization of the nonlinear model around the current state estimate [56]. In this way, WSLR is more accurate in the statistical sense than first-order linearization, which does not account for the "probabilistic spread" at the point of linearization. By representing the forward dynamics in terms of WSLR, a backward information filter can be formulated that does not require inverting the nonlinear dynamics. Estimates of the forward and backward filter are then optimally combined [29], [52] to generate smoothed estimates in the standard manner. Before presenting the pseudocode for the SPKS, we first give a short review on the relationship between the SPKF and WSLR.

Consider a prior RV $\boldsymbol{x}$ which is propagated through a nonlinear function $g(\boldsymbol{x})$ to obtain a posterior RV $\boldsymbol{z}$. While $\boldsymbol{x}$ is a continuous RV observed at each time index k, for the purpose of explaining the WSLR approach, we temporarily omit the discrete time index k from the state and observation variables. Sigma points $\boldsymbol{\chi}_i, i = 0, 1, \ldots, 2M$ are selected as the prior mean $\bar{\boldsymbol{x}}$ plus and minus the columns of the square root of the prior covariance $\boldsymbol{P_x}$

$$\boldsymbol{\chi} = [\, \bar{\boldsymbol{x}} \quad \bar{\boldsymbol{x}} + \gamma\sqrt{\boldsymbol{P_x}} \quad \bar{\boldsymbol{x}} - \gamma\sqrt{\boldsymbol{P_x}} \,] \tag{18}$$

where M is the RV dimension and γ is the composite scaling parameter. The sigma point set $\boldsymbol{\chi}$ completely capture the mean $\bar{\boldsymbol{x}}$ and the covariance $\boldsymbol{P_x}$ of the prior RV $\boldsymbol{x}$

$$\bar{\boldsymbol{x}} = \sum_{i=0}^{2M} w_i \boldsymbol{\chi}_i \tag{19}$$

$$\boldsymbol{P_x} = \sum_{i=0}^{2M} w_i \left(\boldsymbol{\chi}_i - \bar{\boldsymbol{x}}\right)\left(\boldsymbol{\chi}_i - \bar{\boldsymbol{x}}\right)^T \tag{20}$$

where w_i is a normalized scaler weight for each sigma point. Each prior sigma point is propagated through the nonlinearity to form the posterior sigma point $\boldsymbol{\gamma}_i$

$$\boldsymbol{\gamma}_i = g\left(\boldsymbol{\chi}_i\right) \quad i = 0, 1, \ldots, 2M. \tag{21}$$

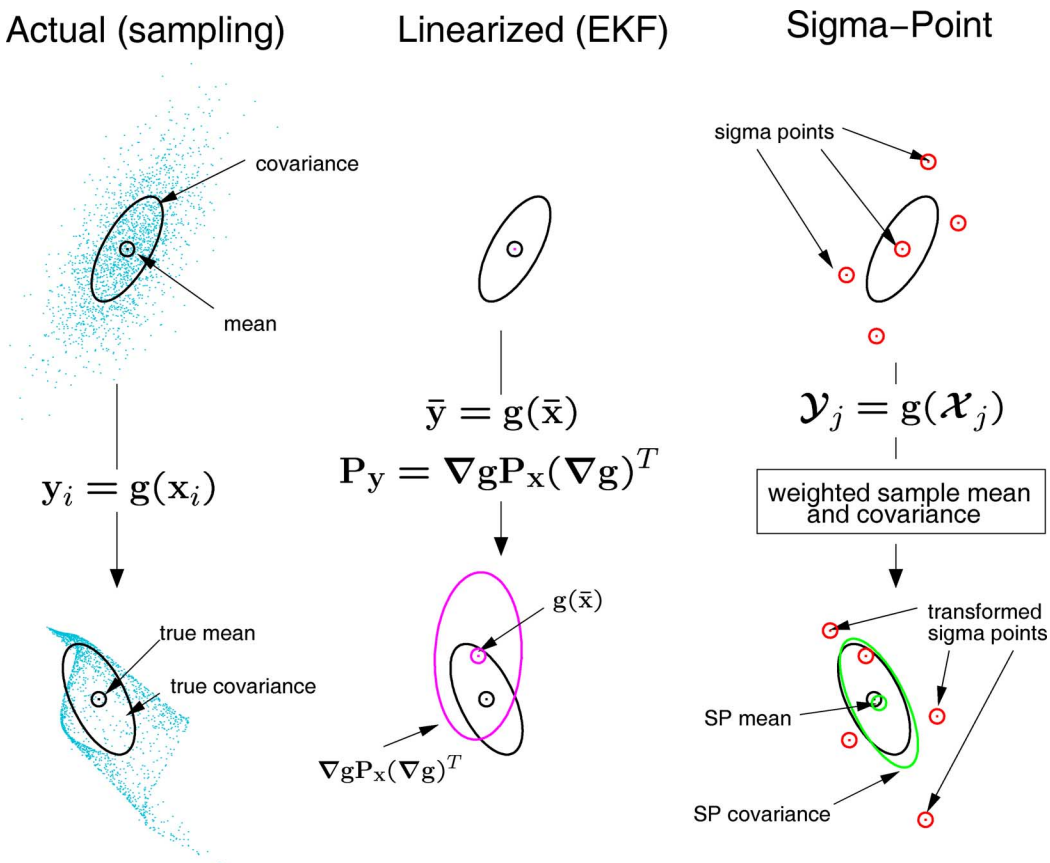


Fig. 3. 2-D example of the sigma-point approach. The accuracy of the sigma-point method in propagating the mean and covariance of the prior GRV through a nonlinear function is compared with Monte Carlo sampling and the EKF approaches.

The posterior statistics can then be calculated using weighted averaging of the posterior sigma points

$$\hat{\boldsymbol{z}} = \sum_{i=0}^{2M} w_i \boldsymbol{\gamma}_i \tag{22}$$

$$\hat{\boldsymbol{P}}_{\boldsymbol{z}} = \sum_{i=0}^{2M} w_i \left(\boldsymbol{\gamma}_i - \hat{\boldsymbol{z}}\right)\left(\boldsymbol{\gamma}_i - \hat{\boldsymbol{z}}\right)^T \tag{23}$$

$$\hat{\boldsymbol{P}}_{\boldsymbol{xz}} = \sum_{i=0}^{2M} w_i \left(\boldsymbol{\chi}_i - \bar{\boldsymbol{x}}\right)\left(\boldsymbol{\gamma}_i - \hat{\boldsymbol{z}}\right)^T. \tag{24}$$

This deceptively simple approach captures the desired posterior statistics more accurately than using standard linearization techniques. The implementation is also simpler, as it avoids the need to analytically linearize the nonlinear function, and only requires direct function evaluations. The performance of the sigma point approach in capturing the mean and covariance of a GRV which undergoes a nonlinear transformation is demonstrated in Fig. 3. The left plot shows the mean and covariance propagation using Monte Carlo sampling. The center plot demonstrates the results using first-order linearization as in the EKF. The right hand plot depicts the performance of the sigma point approach. Note, only 5 sigma-points are needed to approximate the 2-D distribution. Although the same initial conditions and system noises are used, the superior performance of the sigma-point approach is clearly evident.

An alternate view of the sigma point approach can be found by considering the weighted statistical linearization of the nonlinear dynamics

$$\boldsymbol{z} = g\left(\boldsymbol{x}\right) \cong \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{\epsilon} \tag{25}$$

where $\boldsymbol{A}$ and $\boldsymbol{b}$ are the statistical linearization parameters and can be determined by minimizing the expected mean square error which takes into account the uncertainty of the prior RV $\boldsymbol{x}$. Define $J = \mathrm{E}\left[\boldsymbol{\epsilon}^T\boldsymbol{W}\boldsymbol{\epsilon}\right]$ as the expected mean square error with sigma-point weighting matrix $\boldsymbol{W}$

$$\begin{aligned}[\boldsymbol{A}, \boldsymbol{b}] &= \mathrm{argmin} J \\ &= \mathrm{argmin}\left(\mathrm{E}\left[\boldsymbol{\epsilon}^T\boldsymbol{W}\boldsymbol{\epsilon}\right]\right).\end{aligned} \tag{26}$$

The true expectation can be replaced as a finite sample approximation

$$\mathrm{E}\left[\boldsymbol{\epsilon}^T\boldsymbol{W}\boldsymbol{\epsilon}\right] = \sum_{i=0}^{2M} w_i \epsilon_i^T \epsilon_i \tag{27}$$

where the point-wise linearization error $\epsilon_i = \boldsymbol{\gamma}_i - \boldsymbol{A}\boldsymbol{\chi}_i - \boldsymbol{b}$ specifically uses the sigma point selection as above. Now taking partial derivative of J with respect to $\boldsymbol{A}$ and $\boldsymbol{b}$

$$\boldsymbol{A} = \hat{\boldsymbol{P}}_{\boldsymbol{xz}}^T \boldsymbol{P}_{\boldsymbol{x}}^{-1} \tag{28}$$

$$\boldsymbol{b} = \hat{\boldsymbol{z}} - \boldsymbol{A}\bar{\boldsymbol{x}} \tag{29}$$

where the prior mean ($\bar{\boldsymbol{x}}$) and covariance ($\boldsymbol{P}_{\boldsymbol{x}}$) are calculated in (19)–(20) from the prior sigma points. Similarly, the posterior mean ($\hat{\boldsymbol{z}}$) and covariances ($\hat{\boldsymbol{P}}_{\boldsymbol{z}}$ and $\hat{\boldsymbol{P}}_{\boldsymbol{xz}}$) are calculated from the posterior sigma points as shown in (22)–(24). The linearization error $\boldsymbol{\epsilon}$ has zero mean and covariance

$$\boldsymbol{P}_{\boldsymbol{\epsilon}} = \hat{\boldsymbol{P}}_{\boldsymbol{z}} - \boldsymbol{A}\boldsymbol{P}_{\boldsymbol{x}}\boldsymbol{A}^T. \tag{30}$$

From (30)

$$\hat{\boldsymbol{P}}_{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{P}_{\boldsymbol{x}}\boldsymbol{A}^T + \boldsymbol{P}_{\boldsymbol{\epsilon}} \tag{31}$$

we observe that the covariance of the linearization error $\boldsymbol{P}_{\boldsymbol{\epsilon}}$ is added when calculating the posterior covariance $\hat{\boldsymbol{P}}_{\boldsymbol{z}}$. A first-order Taylor series expansion employed by EKF to linearize the nonlinear dynamics neglects this error term. In general, the WSLR technique is an optimal way of linearizing any nonlinear function around the current state estimate [56] in the MMSE sense. The approach explicitly takes into account the prior RV statistics.

To form the SPKF estimator, we consider the nonlinear state space model

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{32}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{n}_k) \tag{33}$$

where $\boldsymbol{x}_k \in \mathbb{R}^M$ is the state, $\boldsymbol{z}_k \in \mathbb{R}^P$ is the observation at time index k, $\boldsymbol{v}_k$, and $\boldsymbol{n}_k$ are Gaussian distributed process and observation noises, $f(.)$ is the nonlinear dynamic model and $h(.)$ is the nonlinear observation model function. The process and observation noise has zero mean and covariances $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ respectively. The SPKF is then derived by recursively applying the sigma point selection scheme at every time index to these dynamic equations (see [51] for more details). Alternatively, the SPKF may be derived using the WSLR from the nonlinear state space

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{f,k}\boldsymbol{x}_k + \boldsymbol{b}_{f,k} + \boldsymbol{G}_{f,k}(\boldsymbol{v}_k + \boldsymbol{\epsilon}_{f,k}) \tag{34}$$

$$\boldsymbol{z}_k = \boldsymbol{A}_{h,k}\boldsymbol{x}_k + \boldsymbol{b}_{h,k} + \boldsymbol{n}_k + \boldsymbol{\epsilon}_{h,k} \tag{35}$$

where $\boldsymbol{A}_{f,k}$, $\boldsymbol{A}_{h,k}$, $\boldsymbol{b}_{f,k}$, $\boldsymbol{b}_{h,k}$ are the statistical linearization parameters and $\boldsymbol{\epsilon}_{f,k}$, $\boldsymbol{\epsilon}_{h,k}$ are the linearization errors with mean zero and covariance $\boldsymbol{P}_{\boldsymbol{\epsilon}_f,k}$ and $\boldsymbol{P}_{\boldsymbol{\epsilon}_h,k}$. $\boldsymbol{G}_{f,k}$ is an input matrix that controls the amount of system noise at each state update. All the parameters can be obtained by applying (29) and (28) iteratively at each time index k. Deriving the KF using the linearized state–space results in the exact same equations and estimates as the SPKF (see [56]). The key advantage, however, of the statistically linearized form is that it can be used to derive the necessary backward filter. Omitting the complete derivation (see [44]), the FBSL-SPKS equations, consisting of the forward filter, backward filter, and smoother, are specified in the next sections.

1) Forward Filter: A standard SPKF is used as the forward filter. The task of the SPKF is to estimate $\boldsymbol{x}_k$ at time index k given all past and current measurements. The SPKF recursions, which operates on the nonlinear state space model defined in (32) and (33), are written as follows using the WSLR formulation:

- *Initialization*:

$$\begin{aligned}
\hat{\boldsymbol{x}}_0 &= \mathrm{E}[\boldsymbol{x}_0] \\
\boldsymbol{P}_{\boldsymbol{x}_0} &= \mathrm{E}\left[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0)^T\right] \\
\hat{\boldsymbol{x}}_0^{\mathrm{a}} &= \mathrm{E}[\boldsymbol{x}_0^{\mathrm{a}}] \\
&= [\,\hat{\boldsymbol{x}}_0^T \quad \hat{\boldsymbol{v}}_0^T \quad \hat{\boldsymbol{n}}_0^T\,]^T \\
\boldsymbol{P}_{\boldsymbol{x}_0}^{\mathrm{a}} &= \left[(\boldsymbol{x}_0^{\mathrm{a}} - \hat{\boldsymbol{x}}_0^{\mathrm{a}})(\boldsymbol{x}_0^{\mathrm{a}} - \hat{\boldsymbol{x}}_0^{\mathrm{a}})^T\right] \\
&= \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{x}_0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_0 \end{bmatrix}.
\end{aligned}$$

  For $\mathrm{k} = 1, 2, \ldots, \mathrm{N}$
- *Calculate sigma-points*:

$$\boldsymbol{\chi}_k^{\mathrm{a}} = [\,\hat{\boldsymbol{x}}_k^{\mathrm{a}} \quad \hat{\boldsymbol{x}}_k^{\mathrm{a}} + \Lambda \quad \hat{\boldsymbol{x}}_k^{\mathrm{a}} - \Lambda\,]$$

  where $\Lambda = \sqrt{(L+\lambda)\boldsymbol{P}_{\boldsymbol{x}_k}^{\mathrm{a}}}$ and L is the dimension of the augmented state.
- *Time-update equations*:

$$\begin{aligned}
\boldsymbol{\chi}_{i,k+1|k}^{\boldsymbol{x}} &= \boldsymbol{f}_k\left(\boldsymbol{\chi}_{i,k}^{\boldsymbol{x}}, \boldsymbol{\chi}_{i,k}^{\boldsymbol{v}}\right) i = 0, 1, \ldots, 2L \\
\hat{\boldsymbol{x}}_{k+1}^- &= \sum_{i=0}^{2L} w_i^{(\mathrm{m})} \boldsymbol{\chi}_{i,k+1|k}^{\boldsymbol{x}} \\
\boldsymbol{P}_{\boldsymbol{x}_{k+1}}^- &= \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{ij}^{(\mathrm{c})} \left(\boldsymbol{\chi}_{i,k+1|k}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{k+1}^-\right)\left(\boldsymbol{\chi}_{j,k+1|k}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{k+1}^-\right)^T.
\end{aligned}$$

- *Weighted Statistical Linearization of* $\mathrm{f}(.)$:

$$\begin{aligned}
\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_{k+1}^-} &= \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{ij}^{\mathrm{c}} \left(\boldsymbol{\chi}_{j,k}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_k\right)\left(\boldsymbol{\chi}_{i,k+1|k}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{k+1}^{-1}\right)^T \\
\boldsymbol{A}_{f,k} &= \boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_{k+1}^-}^T \boldsymbol{P}_{\boldsymbol{x}_k}^{-1} \\
\boldsymbol{b}_{f,k} &= \hat{\boldsymbol{x}}_{k+1}^- - \boldsymbol{A}_{f,k}\hat{\boldsymbol{x}}_k \\
\boldsymbol{P}_{\boldsymbol{\epsilon}_f,k} &= \boldsymbol{P}_{\boldsymbol{x}_{k+1}}^- - \boldsymbol{A}_{f,k}\boldsymbol{P}_{\boldsymbol{x}_k}\boldsymbol{A}_{f,k}^T.
\end{aligned}$$

- *Measurement-update equations*:

$$\begin{aligned}
\boldsymbol{\gamma}_{i,k+1|k} &= \boldsymbol{h}_k\left(\boldsymbol{\chi}_{i,k+1|k}^{\boldsymbol{x}}, \boldsymbol{\chi}_{i,k}^{\boldsymbol{n}}\right) i = 0, 1, \ldots, 2L \\
\hat{\boldsymbol{z}}_{k+1}^- &= \sum_{i=0}^{2L} w_i^{(\mathrm{m})} \boldsymbol{\gamma}_{i,k+1|k} \\
\boldsymbol{P}_{\tilde{\boldsymbol{z}}_{k+1}} &= \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{i,j}^{(\mathrm{c})} \left(\boldsymbol{\gamma}_{j,k+1|k} - \hat{\boldsymbol{z}}_{k+1}^-\right) \\
&\quad \cdot \left(\boldsymbol{\gamma}_{i,k+1|k} - \hat{\boldsymbol{z}}_{k+1}^-\right)^T \\
\boldsymbol{P}_{\boldsymbol{x}_{k+1}\boldsymbol{z}_{k+1}} &= \sum_{i=0}^{2L}\sum_{j=0}^{2L} w_{i,j}^{(\mathrm{c})} \left(\boldsymbol{\chi}_{j,k+1|k}^{\boldsymbol{x}} - \hat{\boldsymbol{x}}_{k+1}^-\right) \\
&\quad \cdot \left(\boldsymbol{\gamma}_{i,k+1|k} - \hat{\boldsymbol{z}}_{k+1}^-\right)^T \\
\boldsymbol{K}_{k+1} &= \boldsymbol{P}_{\boldsymbol{x}_{k+1}\boldsymbol{z}_{k+1}} \boldsymbol{P}_{\tilde{\boldsymbol{z}}_{k+1}}^{-1} \\
\hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1}^- + \boldsymbol{K}_{k+1}\left(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1}^-\right) \\
\boldsymbol{P}_{\boldsymbol{x}_{k+1}} &= \boldsymbol{P}_{\boldsymbol{x}_{k+1}}^- - \boldsymbol{K}_{k+1}\boldsymbol{P}_{\tilde{\boldsymbol{z}}_{k+1}}^- \boldsymbol{K}_{k+1}^T.
\end{aligned}$$

TABLE I
SUMMARY OF USER-SPECIFIED PARAMETERS

Name	Symbol	Value
Potential field Force Constant	F_{cr}	10
Integration time	δT	1 sec
Smoothing AR Coefficient	λ	0.95
Number of Access points	M	3-5
Measurement noise variance (foot-switch)	n_k^{f}	10 ft
Measurement noise variance (constrained motion sensor)	n_k^{ms}	5 ft
Measurement noise variance (unconstrained motion sensor)	n_k^{ms}	room dimension
Measurement Lag	L	3
Sigma-Point spread	α	0.85
Sigma-Point weighting term	β	2
Sigma-Point parameter	κ	0
Time spent per calibration point	T_{c}	1 min
Recorded RSSI per calibration point	N_{r}	8-10
RSSI sampling rate	T_{r}	4-8 sec

- *Weighted Statistical Linearization of* $\mathrm{h}(.)$:

$$\begin{aligned}\boldsymbol{A}_{h,k} &= \boldsymbol{P}^T_{\boldsymbol{x}_{k+1}\boldsymbol{z}_{k+1}}\left(\boldsymbol{P}^-_{\boldsymbol{x}_{k+1}}\right)^{-1}\\ \boldsymbol{b}_{h,k} &= \hat{\boldsymbol{z}}^-_{k+1} - \boldsymbol{A}_{h,k}\hat{\boldsymbol{x}}^-_{k+1}\\ \boldsymbol{P}_{\boldsymbol{\epsilon}_h,k} &= \boldsymbol{P}_{\hat{\boldsymbol{z}}_{k+1}} - \boldsymbol{A}_{h,k}\boldsymbol{P}^-_{\boldsymbol{x}_{k+1}}\boldsymbol{A}^T_{h,k}\end{aligned}$$

- where

$$\begin{aligned}\boldsymbol{x}^{\mathrm{a}} &= [\,\boldsymbol{x}^T \quad \boldsymbol{v}^T \quad \boldsymbol{n}^T\,]^T\\ \boldsymbol{\chi}^{\mathrm{a}} &= [\,(\boldsymbol{\chi}^{\boldsymbol{x}})^T \quad (\boldsymbol{\chi}^{\boldsymbol{v}})^T \quad (\boldsymbol{\chi}^{\boldsymbol{n}})^T\,]^T\\ \boldsymbol{P}^{\mathrm{a}}_{\boldsymbol{x}_k} &= \begin{bmatrix}\boldsymbol{P}_{\boldsymbol{x}_k} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Q}_k & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_k\end{bmatrix}.\end{aligned}$$

- *Parameters*:
  λ is the composite scaling parameter

$$\lambda = \alpha^2(L+\kappa) - L$$

and $w_i^{(\mathrm{c})}$ and $w_i^{(\mathrm{m})}$ are the scaler sigma-point weights defined as [56]

$$\begin{aligned}w_0^{(\mathrm{c})} &= \frac{\lambda}{(L+\lambda)} + \left(1-\alpha^2+\beta\right), i=0\\ w_0^{(\mathrm{m})} &= \frac{\lambda}{(L+\lambda)}, i=0\\ w_i^{(\mathrm{c})} &= \frac{1}{2(L+\lambda)}, i=1,2,\ldots,2L\\ w_i^{(\mathrm{m})} &= \frac{1}{2(L+\lambda)}, i=1,2,\ldots,2L\end{aligned}$$

where α controls the size of the sigma point spread and should be within $0 \le \alpha \le 1$ to avoid sampling non-local points when the nonlinearities are strong [56]. $\beta \ge 0$ is the weighting term which incorporates the higher order moments of the prior distribution. The sigma point approach can effectively capture the first two moments (mean and covariance) of the distribution. The parameter β also can be used to minimize the error in approximating the higher order moments of the distribution. For a Gaussian prior, we set $\beta = 2$[51]. The parameter κ is used to ensure the positive definiteness of the covariance estimates. Setting $\kappa \ge 0$ should work for most cases. The value of all user determined parameters are summarized in Table I. $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are the process and observation noise covariances. Note that the measurement-update equations may be skipped when observations are unavailable. This allows for multi-rate processing in which the state estimates are updated at a higher rate than the sensor sampling or to accommodate missing observations.

2) Backward Filter: An information filter is used to estimate the states in the backward direction given all the present and future measurements. The backward filter recursion is derived from the statistically linearized state space computed during the forward pass (see [44]). The pseudocode is given as follows:

- Initializations:

$$\begin{aligned}\boldsymbol{S}_{\mathrm{N}+1} &= 0\\ \hat{\boldsymbol{y}}_{\mathrm{N}+1} &= 0\end{aligned}$$

where $\boldsymbol{S}_k = (\boldsymbol{P}_{\boldsymbol{x}_k})^{-1}$ is the information matrix and $\hat{\boldsymbol{y}}_k = \boldsymbol{S}_k\hat{\boldsymbol{x}}_k$ is defined as the information state.

- Time-update equations:

$$\begin{aligned}\boldsymbol{S}^-_k &= \boldsymbol{A}^T_{f,k}\boldsymbol{S}_{k+1}\boldsymbol{A}_{f,k} - \boldsymbol{A}^T_{f,k}\boldsymbol{S}_{k+1}\boldsymbol{G}_{f,k}\\ &\quad\cdot\left[\left(\boldsymbol{P}_{\epsilon_f,k}+\boldsymbol{Q}_k\right)^{-1} + \boldsymbol{G}^T_{f,k}\boldsymbol{S}_{k+1}\boldsymbol{G}_{f,k}\right]^{-1}\\ &\quad\cdot\boldsymbol{G}^T_{f,k}\boldsymbol{S}_{k+1}\boldsymbol{A}_{f,k}.\end{aligned}$$

Define $\boldsymbol{K}_{b,k}$ as the backward gain matrix

$$\boldsymbol{K}_{b,k} = \boldsymbol{S}_{k+1}\boldsymbol{G}_{f,k}\left[\left(\boldsymbol{P}_{\epsilon_f,k}+\boldsymbol{Q}_k\right)^{-1} + \boldsymbol{G}^T_{f,k}\boldsymbol{S}_{k+1}\boldsymbol{G}_{f,k}\right]^{-1}.$$

Then substituting $\boldsymbol{K}_{b,k}$ on $\boldsymbol{S}^-_k$,

$$\begin{aligned}\boldsymbol{S}^-_k &= \boldsymbol{A}^T_{f,k}\left(\boldsymbol{I}-\boldsymbol{K}_{b,k}\boldsymbol{G}^T_{f,k}\right)\boldsymbol{S}_{k+1}\boldsymbol{A}_{f,k}\\ \hat{\boldsymbol{y}}^-_k &= \boldsymbol{A}^T_{f,k}\left(\boldsymbol{I}-\boldsymbol{K}_{b,k}\boldsymbol{G}^T_{f,k}\right)\left(\hat{\boldsymbol{y}}_{k+1}-\boldsymbol{S}_{k+1}\boldsymbol{b}_{f,k}\right)\end{aligned}$$

- Measurement-update equations:

$$\begin{aligned}\boldsymbol{S}_k &= \boldsymbol{S}^-_k + \boldsymbol{A}^T_{h,k}\left(\boldsymbol{P}_{\epsilon_h,k}+\boldsymbol{R}_k\right)^{-1}\boldsymbol{A}_{h,k}\\ \boldsymbol{e}_k &= (\boldsymbol{z}_k - \boldsymbol{b}_{h,k})\\ \hat{\boldsymbol{y}}_k &= \hat{\boldsymbol{y}}^-_k + \boldsymbol{A}_{h,k}\left(\boldsymbol{P}_{\epsilon_h,k}+\boldsymbol{R}_k\right)^{-1}\boldsymbol{e}_k.\end{aligned}$$

Note that as the WSLR state space is different than the standard linear state-space used by the Kalman filter, the resulting time and measurement update equations differ from standard backward Kalman equations. Specifically, note how the correction terms $\boldsymbol{P}_{\epsilon_f,k}$ and $\boldsymbol{P}_{\epsilon_h,k}$ are fed back into the time-update and measurement-update equations. This term is absent in the standard information filter formulation [29], [52]. The more severe the nonlinearity is over the uncertainty region of the state, the higher will be the linearization error covariance matrices. This correction term appears due to the statistical linearization as it considers the covariance of the prior RV while linearizing the nonlinear model. A first-order Taylor series expansion is less accurate because it does not consider this error term.

3) Smoothing: As the last step in the estimation process, the forward and backward filter estimates are optimally combined to form the smoothed estimates

$$\begin{aligned}\boldsymbol{P}^s_k &= \left[\left(\boldsymbol{P}_{\boldsymbol{x}_k}\right)^{-1} + \boldsymbol{S}^-_k\right]^{-1}\\ \hat{\boldsymbol{x}}^s_k &= \left(\boldsymbol{I}+\boldsymbol{P}_{\boldsymbol{x}_k}\boldsymbol{S}^-_k\right)^{-1}\hat{\boldsymbol{x}}_k + \boldsymbol{P}^s_k\hat{\boldsymbol{y}}^-_k.\end{aligned}$$

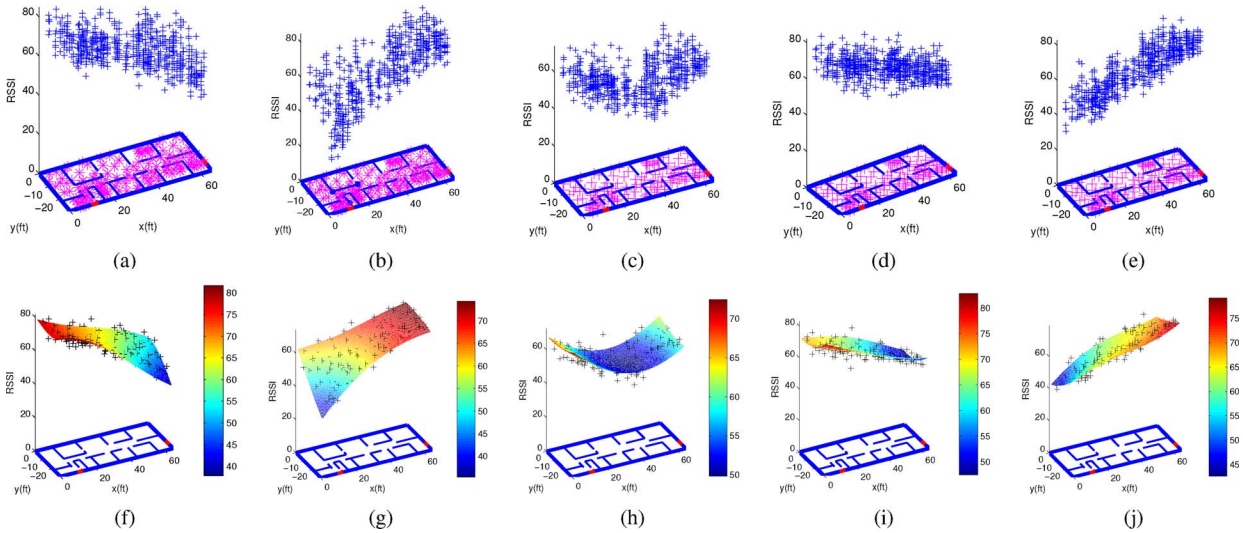


Fig. 4. (a)–(e): raw RSSI values from five access points collected during calibration at Point of Care test lab-I, (f)–(j): fitted RBF maps.

For tracking purposes, the FBSL-SPKS provides an offline estimate of the position and velocity trajectories after all data up to time N has been collected. Alternatively, pseudo real-time estimates may be achieved by dividing the data into blocks (e.g., $N = \sum N_i$) and then performing the forward–backward operation on the buffered blocks of data as they become available.

B. Fixed Lag Sigma-Point Kalman Smoother (FL-SPKS)

In FL-SPKS, the objective is to estimate the current state using all the past, present, and L future measurements, where L is the fixed lag. Alternatively, this may be viewed as estimating the lagged state $\boldsymbol{x}_{k-L}$ given all measurements up to the current time k. The FL-SPKS is specified by simply defining a new augmented state space

$$\tilde{\boldsymbol{x}}_{k+1} = \begin{bmatrix} \boldsymbol{x}_{k+1} & \boldsymbol{x}_k & \cdots & \boldsymbol{x}_{k-L} \end{bmatrix}^T \tag{36}$$

$$= \begin{bmatrix} \boldsymbol{f}_k(\boldsymbol{x}_k) \\ \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \tilde{\boldsymbol{x}}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{I} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \boldsymbol{v}_k \tag{37}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{n}_k. \tag{38}$$

The standard SPKF recursions shown in Section III-A1 are applied directly to the augmented system. The fixed-lag estimate of the last element of the augmented state vector $\boldsymbol{x}_{k-L}$ will be equal to the $\hat{\boldsymbol{x}}_{k-L}$ given measurements up to and including time index k. The increased state dimension increases the overall computational complexity of the algorithm. However, the FL-SPKS provides sequential estimates of the states delayed by L measurements, and is thus more appropriate to online implementation than the FBSL-SPKS.

IV. Experimental Results

Implementation and testing was performed at several "living-laboratories" (also called Point-of-Care labs) used to develop monitoring and assistive technologies for the elderly. A number of trials were conducted in which different subjects followed a predefined path. While walking, the subject periodically noted down the ground truth. The Ekahau real-time positional engine was also turned on during these tests. Although the exact location estimation algorithm used by the Ekahau software was not known, this still provides a commercial benchmark for evaluation of our approach. Note that the same calibration data was used for both the SPKS based tracker and the Ekahau's positioning engine. The performances of the SPKS-based tracker and the Ekahau real time tracking engine displayed in this paper were performed at three different sites.

A. Test Lab-I

The Point-of-care test lab-I was setup with five access points located at the four corners and at the center. The size of the test lab-I is 60 ft by 30 ft. In the entire environment, calibration was performed first in order to measure RSSI variability emanating from each access point. The floor plan was divided into $P = 15$ sections. Each room was considered a section and the long corridors were divided into multiple sections. In each section, nine points were chosen to perform calibration in such a way that eight points formed the periphery of an octagon and the remaining point was at the center of the octagon. At each calibration point, a person carrying a RSSI tag spends around one minute to collect the training RSSI data. Roughly 8–10 RSSI measurements are recorded at each calibration points for a total of 1065 measurements.

As described earlier, an *RBF* network is used to fit a nonlinear map between known calibration locations and the collected RSSI values. The raw RSSI at each calibration point and the RSSI calibrated maps for five access points were shown in Fig. 4(a)-(j).

We conducted several trials of moving test in which subjects walked at a normal speed following a predefined path. Two different subjects were used as RSSI variability is observed to be subject dependent. In one trial, subject 1 took 174 seconds to complete the path and 25 RSSI observations were recorded during that time period. Subject 2 completed

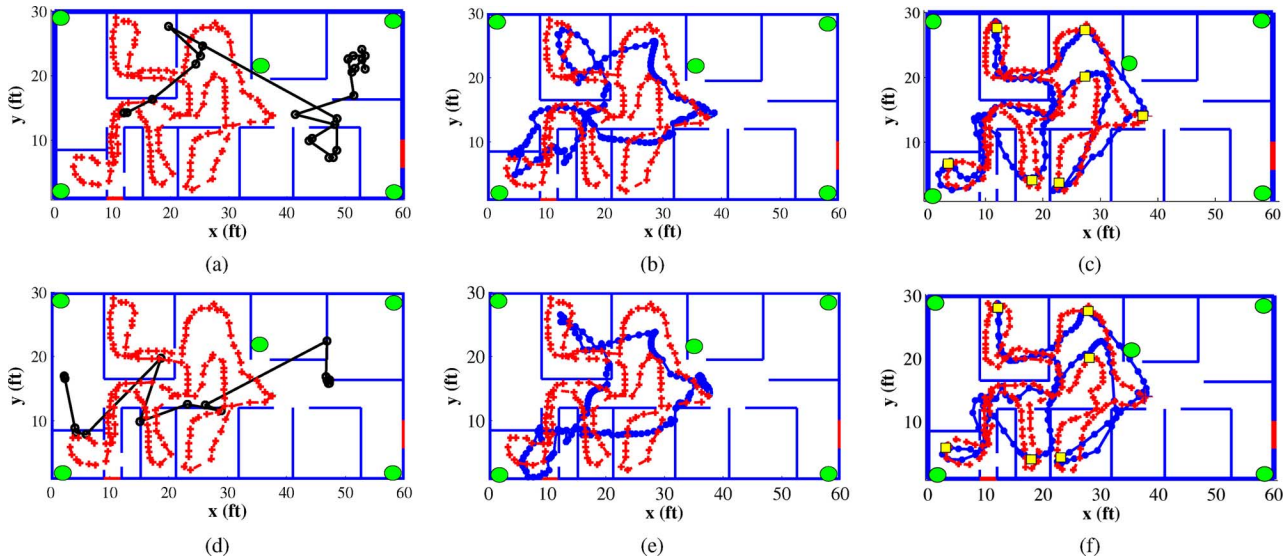


Fig. 5. Tracking performance in test lab-I, (a) and (d) Ekahau estimates (red/light gray: ground truth, black: estimate), (b) and (e) SPKS estimates using RSSI measurements (red/light gray: ground truth, blue/dark gray: SPKS estimate), (c) and (f) SPKS estimates using RSSI + foot switch observations (red/light gray: ground truth, blue/dark gray: SPKS estimate). Yellow/light gray rectangular boxes indicate the position of the foot switches on the floor plan. The position of the access points are shown by green/gray circles on the floorplan. First row is for subject 1 and the second row is for subject 2.

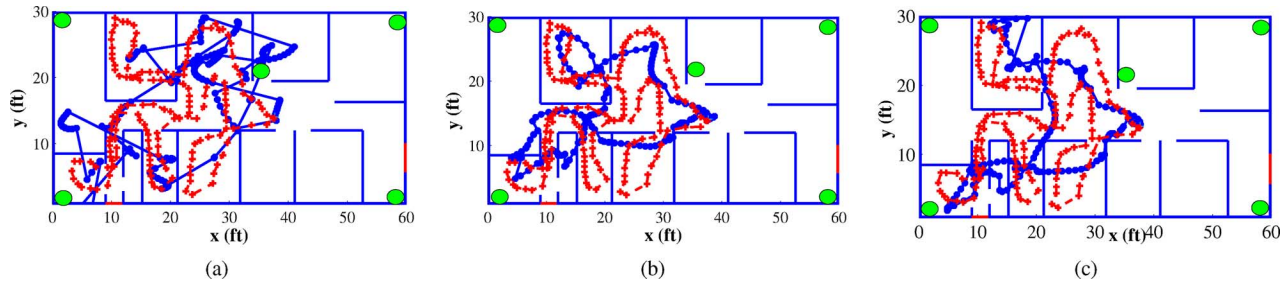


Fig. 6. (a) SPKF estimates, (b) FBSL-SPKS estimates, (c) FL-SPKS estimates (red/light gray: ground truth, blue/dark gray: SPKS estimates). The position of the access points are shown by green/gray circles on the floorplan. All estimates used only RSSI sensors.

the same path in 164 seconds and recorded 19 RSSI observations. The sampling rate varied between 4–8 s during tracking. Multi-rate filtering was implemented so that the time-update equations still provide estimates of the position and velocity at every second. Approximate ground truth was collected periodically during the walking and is also shown in the plots. Fig. 5(a)–(f) compares the estimates obtained from the Ekahau engine and the SPKS tracker. From Fig. 5(a) and (d), it can be seen that the estimates from Ekahau tracking engine are very inaccurate and often fails to even localize the person in the correct region/room. The SPKS tracker with RSSI only observations clearly tracked the person with greater accuracy [see Fig. 5(b) and (e)].

Only the FBSL-SPKS estimates are shown in the previous figures. The tracking performance of SPKF, FBSL-SPKS and FL-SPKS are compared in Fig. 6(a)-(c). As shown, both the SPKS estimates are superior to the SPKF. In addition, the FL-SPKS estimates are slightly less accurate compared to the FBSL-SPKS estimates.

When RSSI observations were integrated with foot-switch signals, the accuracy of the SPKS based tracker improved even further [Fig. 5(c) and (f)]. Note that we set the variance of the foot-switch sensors to be 10 ft in our experiments.

B. *Test Lab-II*

Similar to POCL test lab-I, calibration was performed in the other POCL test lab for each of the five Wi-Fi access points. The lab is also fitted with a number of IR motion sensors instead of foot switches. There are two variants of motion sensors installed in the houses based on their field-of-view. The full beam width unconstrained sensors are generally installed one per room and have variability that matches the full dimension of the room. The constrained sensors have limited beam width and are generally installed along corridors. The variability of the constrained sensors is thus significantly lower than the unconstrained. In Fig. 7(a)–(c), we demonstrate a walking experiment comparing the Ekahau performance to the SPKS tracker. The Ekahau estimates as observed in Fig. 7(a) are mostly stuck in one portion of the house. The SPKS tracker performance using RSSI, with and without IR motion sensors, are depicted in Fig. 7(b) and (c). While still superior to the Ekahau estimates, the small size of the POCL lab-II (30 ft by 22.5 ft) and the existence of a large number of furniture limited the performance of the SPKS tracker compared to lab-I. Adding the motion sensors with RSSI improved the SPKS tracker accuracy in spite of the high false alarm and large variability of the motion sensors.

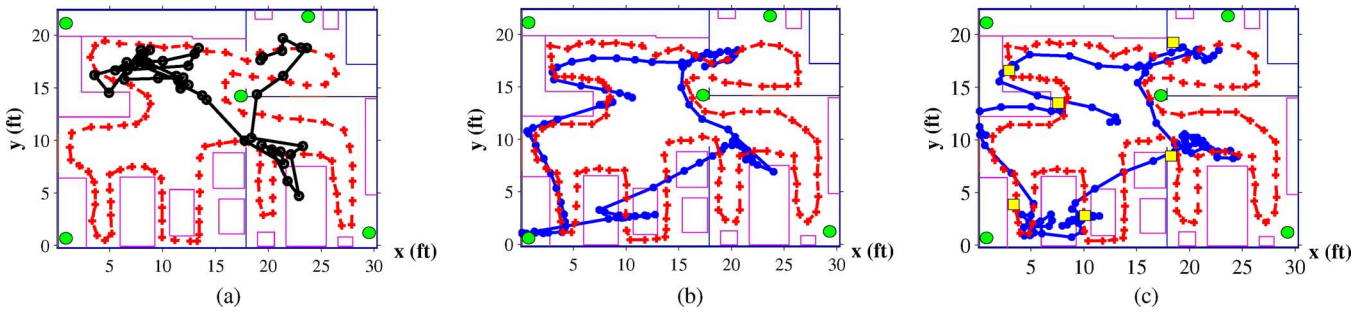


Fig. 7. Tracking performance in test lab-II, (a) Ekahau estimates (red/light gray: ground truth, black: Ekahau estimate), (b) SPKS estimates using RSSI measurements (red/light gray: ground truth, blue/dark gray: SPKS estimate), (c) SPKS estimates using RSSI + foot switch observations (red/light gray: ground truth, blue/dark gray: SPKS estimate). Small yellow/light gray rectangular boxes indicate the location of the motion sensors on the floorplan. The position of the access points are shown by green/gray circles on the floorplan. The furniture positions are shown as magenta/white rectangular boxes.

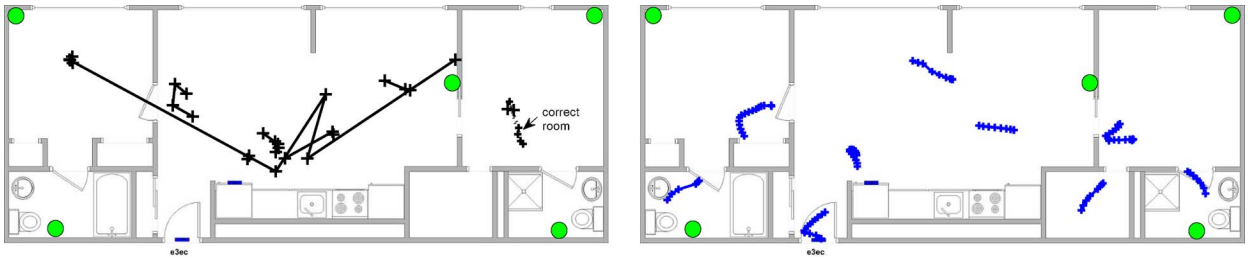


Fig. 8. Tracking performance in test lab-III, (a) Ekahau estimates (brown/dark gray: Ekahau estimate), (b) SPKS estimates using RSSI only measurements (blue/dark gray: SPKS estimate). The position of the access points are shown by green/gray circles on the floorplan.

C. Test Lab-III

In Fig. 8(a)-(b), we demonstrate an additional moving test at a third location. The size of the lab-III is 55 ft by 25 ft. Similar to the test lab I and II, this living lab is also equipped with five wireless access points placed at four corners and at the center. The entire floorplan is divided into nine sections. In each section, an octagonal grid is formed to perform calibration. Due to a problem with the RSSI tags, a limited amount of calibration data was collected (only 212 RSSI measurements for the entire house). We would thus expect worse tracking performance corresponding to only room level localization accuracy. During the moving test, a subject walked on the calibration grid points at each section along a counter clockwise direction. Fig. 8(a) shows the Ekahau localization performance. As seen in Fig. 8(a)), the Ekahau tracking engine failed to localize the person in the correct section except for a single case [as marked in Fig. 8(a)]. Most of the Ekahau estimates are randomly centered around the middle of the entire floor plan. However, the proposed FBSL-SPKS based tracker correctly localized the person in all of the sections as observed in Fig. 8(b).

V. Conclusion

A new method and system has been developed for RSSI based indoor localization and tracking. Instead of using simple fingerprinting or a fixed *a priori* distribution for the RSSI tags, an observation function is generated from RSSI calibration data by fitting nonlinear maps between known calibration locations and RSSI mean values. The RSSI maps are incorporated into a Bayesian framework that fuses all sensor measurements with a simple dynamic model of walking. The dynamic model consists of a random walk model augmented with repulsive forces to account for room–wall configurations. For the Bayesian inference, we use sigma point Kalman filters (SPKF), which provide improved performance over standard extended Kalman filters (EKF) while maintaining the computational efficiency. We further developed two sigma-point Kalman smoother (SPKS) based implementations (forward–backward and fixed-lag) that provide considerable improvement in tracking accuracy compared with the standard SPKF. The SPKS tracker can accommodate multi-rate processing where state estimates are determined at a higher rate (e.g., every second) while RSSI observations occur at slower update rate. Missing observations are also easily handled by the approach.

While the primary sensors are Wi-Fi tags, the approach can also incorporate multiple types of sensors. In the current implementation, both IR motion sensors and simple foot-switches were incorporated. Table II summarizes the performance and superiority of the proposed SPKS based tracker over other popular estimation techniques in terms of root-mean-square position error (RMSE). The trials used in Table II were performed in a number of different living laboratories. As a predominantly software solution, the approach provides the flexibility to incorporate sensors from multiple manufacturers. Performance was evaluated in a number of "living laboratories," where tracking accuracy was demonstrated to be superior to the available industry positioning engine developed by Ekahau, Inc. The proposed system is currently being deployed into a number of houses in order to continuously monitor elderly for clinical purposes.

TABLE II
PERFORMANCE COMPARISON OF THE SPKS WITH OTHER ESTIMATORS. OVERALL RMSE IS CALCULATED RELATIVE TO THE OBSERVED TRUE TRAJECTORY OVER 20 DIFFERENT TRIALS PERFORMED AT VARIOUS TEST LABS AND REAL HOUSES. RMSE FOR TEST LAB-III IS NOT SHOWN AS AN ACCURATE GROUND TRUTH WAS NOT AVAILABLE

Estimator	**Overall** RMSE (ft)	RMSE**:(Test Lab-I)** (ft)	RMSE**:(Test Lab-II)** (ft)
Ekahau (RSSI)	23.88	28.4	20.22
EKF(RSSI)	10.32	9.73	12.31
EKS(RSSI)	8.54	7.49	10.2
SPKF(RSSI)	8.95	8.21	9.85
SPKS(RSSI)	6.45	5.68	8.38
SPKS(RSSI+IR motion sensor)	5.92	N/A	6.26
SPKS(RSSI+footswitch)	1.44	1.44	N/A

Although the SPKS-based approach is capable of accurate tracking, several factors may limit the performance of the proposed approach. RSSI noise, spatial variability, and sampling rate all affect accuracy. Complexity of the position-RSSI correspondence due to severe multipath, NLOS propagation, metal reflection, interference from other Wi-Fi devices, and RSSI noise can pose significant challenges in successful generation of the observation maps. The accuracy of the RSSI observation maps also depend on the number of RSSI measurements collected over the entire indoor environment during calibration. A sufficient number of calibration points at different locations must be collected to capture the full spatial characteristics of the environment. The octagonal positioning used in this study may or may not be sufficient. While a complex spatial variability in the RSSI makes fitting the observation maps more difficult, the complexity of the maps is precisely what makes tracking possible. A relative flat observation map is non-informative. The actual shape of the RSSI map is also influenced by the type, location, and orientation of the RF access points, which must be properly located and tuned so that RSSI can be acquired over the entire house for each access point. During tracking, accuracy is further influenced by the RSSI noise and sampling rate, which both contributes to location uncertainty. The 4–8 s sampling rate in this study was due to hardware limitations. Clearly, a faster sampling rate is always preferred. Incorporation of non-RSSI sensors may also affect accuracy. In this paper, motion sensors were used. However, we only incorporated a very simply model that did not account for beam patterns, false alarms, or latency.

Additional planned research includes refined models of walking motion and better likelihood models for the IR motion sensors. We also plan to investigate self calibration (i.e., simultaneous localization and mapping), whereby the parameters of the RSSI maps are continuously updated to account for changes in the environment or to even avoid the initial offline calibration procedure. Wall mounted sonar range sensors are also being investigated that would provide an alternative to RSSI, allowing for unobtrusive localization without the use of body worn tags.

ACKNOWLEDGMENT

The authors would like to thank M. Pavel and T. Hayes for project support and use of laboratory facilities, and A. Tsay, E. Earl and J. Yeargers for help with software support and data collection.

REFERENCES

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 40, no. 1, pp. 91–102, Jan. 1992.
[2] P. Bahl and V. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE Infocomm'00*, Apr. 2000, pp. 775–784.
[3] N. Priyantha, A. Charraborty, and H. Balakrishnan, "The cricket location-support system," in *Proc. ACM Mobicomm'00*, Jul. 2000, pp. 32–43.
[4] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitous computing: A case study," in *Proc. Int. Conf. Ubiquitous Comput. (UBICOMP)*, 2004, pp. 88–106.
[5] [Online]. Available: http://www.sonitor.com/
[6] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert 5, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Proc. Int. Conf. Pervasive Comput. (Pervasive)*, 2005, pp. 116–133.
[7] Y. Cheng, Y. Chawathe, A. LaMraca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *Proc. 3rd Int. Conf. Mobile Syst., Applicat., Services*, 2005, pp. 233–245.
[8] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proc. 3rd Int. Conf. Mobile Syst., Applicat., Services*, 2005, pp. 205–218.
[9] [Online]. Available: http://www.ekahau.com/
[10] [Online]. Available: http://www.ubisense.net/
[11] X. Li and K. Pahlavan, "Super-resolution TOA estimation with diversity for indoor geolocation," *IEEE Trans. Wireless Communications*, vol. 3, pp. 224–234, 2004.
[12] A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 689–705, Jun. 2007.
[13] J. Pan, J. Kwok, Q. Yang, and Y. Chen, "Accurate and low-cost location estimation using kernels," in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 1366–1370.
[14] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioing systems based on location fingerprinting," in *Proc. INFOCOM*, 2004, vol. 2, pp. 1012–1022.
[15] T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen, "A probabilistic approach to WLAN user location estimation," *Int. J. Wireless Inf. Netw.* , vol. 9, no. 3, pp. 155–164, 2002.
[16] A. Ladd, K. Bekris, A. Rudys, G. Marceau, L. Kavraki, and D. Wallach, "Robotics based location sensing using wireless Ethernet," in *Proc. Mobicom*, 2002, pp. 227–238.
[17] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A wireless lan-based indoor positioning technology," *IBM J. Res. Develop.*, vol. 48, no. 5/6, pp. 617–626, 2004.
[18] R. Singh, L. Macchi, C. Regazzoni, and K. Plataniotis, "A statistical modelling based location determination method using fusion in wlan," in *Proc. Int. Workshop Wireless Ad-Hoc Networks*, 2005.
[19] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[20] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Comput.*, vol. 2, no. 3, pp. 24–33, 2003.
[21] A. Elfes, E. Prassler, and J. Scholz, "Tracking people in a railway station during rush hour," in *Proc. Comput. Vis. Syst. ICVS'99*, 1999, pp. 162–179.
[22] A. Fod, A. Howard, and M. Mataric, "Laser-based people tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 3024–3029.

[23] C.-T. Hsieh, C.-Y. Chen, and Y.-K. Wu, "People tracking system using Kalman filter," in *Proc. Signal Image Process.*, 2002.
[24] J. Krumm and E. Horvitz, "Locadio:inferring motion and location from Wi-Fi signal strengths," in *Proc. Int. Conf.Mobile Ubiquitous Syst. :Netw. Services (MobiQuitous 04)*, 2004, pp. 4–14.
[25] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–435, Feb. 2002.
[26] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proc. Robot.: Sci. Syst.*, 2006.
[27] J. Letchner, D. Fox, and A. LaMarce, "Large-scale localization from wireless signal strength," in *Proc. National Conf. Artif. Intell. (AAAI-05)*, 2005, pp. 15–20.
[28] D. Hhnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with RFID technology," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2004, pp. 1015–1020.
[29] F. Lewis, *Optimal Estimation with an Introduction to Stochastic Control Theory*. New York: Wiley, 1986.
[30] Q. Meng, Y. Sun, and Z. Cao, "Adaptive extended Kalman filter (aekf)-based mobile robot localization using sonar," *Robotica*, vol. 18, pp. 459–473, 2000.
[31] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.
[32] S. Seidal and T. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Trans. Antennas Propag.*, vol. 40, no. 2, pp. 207–217, Feb. 1992.
[33] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," in *Proc. Int. Conf. Ubiquitous Computing (UbiComp)*, 2005, pp. 87–104.
[34] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "Gpps: A Gaussian process positioning system for cellular networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003.
[35] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. Symp. 2000 Adaptive Syst. Signal Process., Commun., Control (AS-SPCC)*, Oct. 2000, pp. 153–158.
[36] E. Wan and R. van der Merwe, *Kalman Filtering and Neural Networks*, 1st ed. New York: Wiley, 2001, ch. 7, pp. 221–280.
[37] R. van der Merwe and E. Wan, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," in *Proc. Workshop Adv. Mach. Learn.*, Jun. 2003.
[38] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. Amer. Control Conf.*, 1995, pp. 1628–1632.
[39] M. Norgaard, N. Poulsen, and O. Ravn, "New developments in state estimation for nonlinear systems," *Automatica*, vol. 36, pp. 1627–1638, Nov. 2000.
[40] R. van der Merwe and E. A. Wan, "The square-root unscented Kalman filter for state and parameter estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Salt Lake City, UT, May 2001, vol. 6, pp. 3461–3464.
[41] R. van der Merwe and E. Wan, "Sigma-point Kalman filters for integrated navigation," in *Proc. 60th Ann. Meeting Inst. Nav. (ION)*, Jun. 2004.
[42] J. Crassidis, "Sigma-point Kalman filtering for integrated gps and inertial navigation," in *Proc. AIAA Guidance, Nav., Control Conf. Exhibit*, Aug. 2005.
[43] E. Shin, X. Niu, and N. El-Sheimy, "Performance comparison of the extended and the unscented Kalman filter for integrated gps and mems," in *Proc. ION, National Tech. Meeting, NTM*, 2005.
[44] A. S. Paul and E. A. Wan, "A new formulation for nonlinear forward-backward smoothing," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 3621–3624.
[45] A. Paul and E. Wan, "Wi-Fi based indoor localization and tracking using sigma-point kaman filtering methods," in *Proc. IEEE/ION Pos. Loc. Nav. Symp. (PLANS 2008)*, 2008.
[46] X. Li and V. Jilkov, "A survey of maneuvering target tracking-part ii," in *Proc. SPIE Conf. Signal Data Process. Small Targets*, San Diego, CA, Jul.-Aug. 2001, vol. 4473, pp. 559–581.
[47] X. Li and V. Jilkov, "Survey of maneuvering target tracking-part i: Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
[48] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 1179–1187, Sep./Oct. 1989.
[49] S. Haykin, *Neural Networks : A Comprehensive Foundation*. New York: Maxwell Macmillan Int., 1994.
[50] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.*, vol. 82, pp. 35–45, Mar. 1960.
[51] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, pp. 401–422, Mar. 2004.
[52] D. Simon, *Optimal State Estimation*. New York: Wiley Interscience, 2006.
[53] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Autom. Control*, vol. 14, no. 4, pp. 387–390, Aug. 1969.
[54] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA*, vol. 3, no. 8, pp. 1445–1450, 1965.
[55] C. Leondes, J. Peller, and E. Stear, "Nonlinear smoothing theory," *IEEE Trans. Syst. , Sci., Cybern.*, vol. SSC-6, no. 1, pp. 63–71, Jan. 1970.
[56] R. van der Merwe, "Sigma-point Kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, OGI School Sci. and Eng., Oregon Health Sci. Univ., Beaverton, OR, 2004.

Anindya S. Paul (S'02) received the B.Eng. degree from the Sikkim Manipal Institute of Technology, Manipal, India, in 2001 the M.S. degree in electrical engineering from Wright State University (WSU), Dayton, OH, in 2003where he collaborated with the Wright Patterson Air Force Base for his research work. He is currently pursuing the Ph.D. degree from the OGI School of Science and Engineering, Oregon Health and Science University (OHSU), Beaverton. His doctoral dissertation is on " Sigma-Point Kalman Filtering: Algorithm and analysis with applications to indoor localization and tracking," which he expects to complete in 2009.

In 2003, he accepted the offer to work as a Graduate Research Assistant for the Adaptive Systems Lab, Computer Science and Electrical Engineering, Department, OGI School of Science and Engineering, OHSU. His broad research interests lies in the fields of probabilistic inference and Bayesian learning, statistical pattern recognition, artificial neural networks and machine learning. His research work deals with unobtrusive indoor localization and tracking using Wi-Fi received signal strength identification (RSSI) and ultrasonic sonar range sensors, developing new formulation for sigma-point Kalman smoothers for nonlinear systems, analyzing stability and convergence criterions for sigma-point Kalman filter, and developing a dual-estimation framework (SLAM) for autonomous navigation of a vehicle in unknown environment.

Mr. Paul received "Special Award for Collaborative Achievements" for research in multiple collaborative projects and research groups in 2007 from OHSU.

Eric A. Wan (M'91) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1987, 1988, and 1994, respectively.

He joined the faculty at the OGI School of Science and Engineering, Oregon Health and Science University (OHSU), Beaverton, in 1994. He is currently an Associate Professor in the Division of Biomedical Engineering, OHSU. His research is in algorithms and architectures for machine learning and adaptive signal processing. He has led a diverse number of research projects over the years ranging from autonomous unmanned aerial vehicles to speech enhancement for cellular communications. He is best known for his work on estimation and probabilistic inference, and in particular the development of the sigma-point Kalman filter/unscented Kalman filter. Recent research has focused on applications of machine learning to biomedical engineering, with projects including noninvasive glucose monitoring, real-time classification of flying insects for diseases vector monitoring, and localization and tracking of elderly in assisted living homes. He holds several patents in adaptive signal processing and has authored over 60 technical papers in the field.

Dr Wan is a member of the INNS, Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.