WI-FI BASED INDOOR LOCALIZATION AND TRACKING USING SIGMA-POINT KALMAN FILTERING METHODS

Anindya S. Paul and Eric A. Wan

OGI school of Science and Engineering
Oregon Health and Science University (OHSU)
20000 NW Walker Road, Beaverton, OR 97006, U.S.A.

ABSTRACT

Estimating the location of people and tracking them in an indoor environment poses a fundamental challenge in ubiquitous computing. The accuracy of explicit positioning sensors such as GPS is often limited for indoor environments. In this study, 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. At the core of our system is a novel location and tracking algorithm using a sigma-point Kalman smoother (SPKS) based Bayesian inference approach. The proposed SPKS fuses a predictive model of human walking with a number of low-cost sensors to track 2D position and velocity. Available sensors include Wi-Fi received signal strength indication (RSSI), binary infrared (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— sigma-point Kalman filter, sigma-point Kalman smoother, indoor tracking, RSSI tracking

1. INTRODUCTION

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 GPS work worldwide and can sometimes achieve centimeter accuracy. However, GPS requires a direct view to several satellites, resulting in limited performance for indoor environments. Hence there remains a need to develop new location sensing technologies based on both existing opportunistic 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 and research prototype systems currently exist for indoor localization. Systems typically use infra-red (IR), radio-frequency (RF), or ultra-sound sensors [1–4]. Although they show the necessary potential in indoor tracking, each has their own limitations. The Active Badge System is one example of an early location-aware application that determines the location of an individual by use of an ”active badge” that emits an unique IR code [1]. A network of sensors pick up the periodic IR waveforms to triangulate the individual. Poor IR scalability and high maintenance overheads were some of the drawbacks faced by this system. Two popular systems, RADAR [2] and Cricket [3], use RF, and RF with ultrasonic sensors respectively. RADAR compares the user’s RSSI observation with a set of prestored signal strength measurements known as ”fingerprints” at each of the base stations to identify the user’s coordinates. RADAR uses a *k-nearest-neighbors* method in order to find the closest match between known fingerprints and observed RSSI. The major disadvantages of the fingerprinting method includes 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, metal reflection, and noise. The Cricket system uses time of flight (TOF) difference between RF and ultrasonic pulses in order to localize a person. 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 deploy it in practice. ”Place Lab” evaluates the feasibility of building a large scale location tracking system based on a 802.11 wide-area wireless network [5, 6]. Place Lab uses standard 802.11 access points

Corresponding Author: Anindya S. Paul
Anindya S. Paul (Ph.D. Candidate) and Eric A. Wan (Associate Professor) are with the Computer Science and Electrical Engineering Department (CSEE), Email Address: anindya@csee.ogi.edu, ericwan@csee.ogi.edu.

and 802.11 radio listener built into the users devices for positioning. The system compares the observed RSSI with a prestored radio map to determine the users position. Although it enjoys the advantage of limited calibration, the accuracy is lower than exiting positioning systems. Ekahau [7] provides a complete tag and software solution, also using RSSI with the 802.11 protocol. 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. There are also a number of software and hardware based systems using ultra wide-band (UWB) and ultrasound frequencies that also use TOF calculations and employ similar methods to that used within GPS receivers [8, 9]. Ubisense [8], for example, can provide high tracking accuracy to within several centimeters, but with significant calibration challenges as well as a very high cost.

Our contribution to indoor location tracking focuses on improving the core estimation algorithms that fuse the various sensor measurements. We take a Bayesian inference approach, which provides a probabilistic framework for combining a predictive model of walking motion with all available sensor observations. Bayesian methods can estimate a person's velocity and acceleration in addition to position, and can also provide a measure of the accuracy of the estimates. A number of variants of probabilistic Bayesian inference approaches have already appeared in the literature [4, 6, 10–18]. In [10, 15], 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 [19]. Meng *et al.* [20] propose an adaptive extended Kalman filter (EKF) algorithm to localize a mobile robot equipped with multiple sonar sensors in an indoor environment. Leonard *et al.* [21] use an EKF approach to accurately localize a mobile robot that relies on the observed distance from the current robot's position to a number of known environmental features. Borthwick *et al.* [22] employ an EKF based technique to track an automated guided vehicle (AGV) using a optical ranger and a known priori map of the environment. Kalman filter and their variants have also been applied to indoor people tracking. For examples, Fod *et al.* [12] and Hsieh *et al.* [13] 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 [4,16–18]. The particle filter based system described by J. 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 only a single Gaussian with fixed manufacturer predefined parameters. However, severe multipath, non-line-of-sight propagation, or metal reflection may cause a highly complex position-RSSI relationship that is not accurately captured by such a simple observation model. Letchner *et al.* [17] introduce a sensor measurement model in the particle filter framework that combines a Wi-Fi signal propagation model [23] and fingerprinting techniques for localization. The methods assumes radially symmetric attenuation of wireless signals and also requires large training data for fingerprinting. Recently, Gaussian processes are also used to generate a measurement likelihood for wireless signal strength measurements in particle filter [16]. However, learning the parameters using graph based hyperparameter estimation can take significant computational resources with slow convergence. The use of hidden Markov modeling (HMM) and grid based approaches for people tracking have also appeared in the literature. The Locadio system [14] use a 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) [24–26]. SPKFs are estimation algorithms, which include the unscented Kalman filter (UKF) [27], central difference Kalman filter (CDKF) [28], and their square-root variants [29]. The SPKF has recently become a popular better alternative to the Extended Kalman Filter (EKF). Additional background on the SPKF will be provided later in this paper. We specifically used our recently proposed forward-backward statistically linearized sigma-point Kalman smoother (FBSL-SPKS) algorithm [30] . For tracking purposes, this is implemented as a fixed interval smoother that use all the past and future measurements to estimate the current state. To make the tracking close to real time, we also develop a fixed lag Kalman smoother (FL-SPKS) algorithm. Both estimators optimally fuse a model of walking motion, room-wall configurations, and all available sensor observations in order to track a person. A random acceleration based human walk model 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. Our observation model use Radial-Basis Function (RBF) networks that provide a nonlinear mapping between known locations and observed RSSI values. These models are fit during a separate calibration process, and thus take into account the various multipath and other room specific characteristics.

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 3 or more standard Wi-Fi

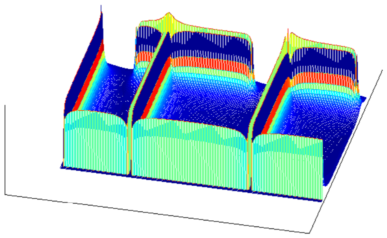


Fig. 1. Room Potential field

access points placed at pre-defined locations. 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 was 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.

2. 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,

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

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

where $\boldsymbol{x}_k$ represent 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. Under a pure Gaussian and linear assumption, the Kalman filter is optimal and provides for an efficient and practical solution. 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 of the specific tracking application.

2.1. Dynamic Model

We define the state vector $\boldsymbol{x} = \begin{bmatrix} x & y & v_x & v_y \end{bmatrix}$, corresponding to 2D position and velocity for tracking purposes. A simple random walk model on the velocities 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 off-line using prior knowledge of wall configurations and large furniture location. Computationally this is achieved by dividing the space into 1 inch square cells. Each cell contains a binary certainty measure $C\left(i,j\right)$ that indicates whether the cell is occupied, *i.e.*, an obstacle exits within the cell. The force $\boldsymbol{F}_{i,j}\left(x,y\right)$ 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.

$$\boldsymbol{F}_{i,j}\left(x,y\right) = -\frac{F_{cr}C\left(i,j\right)}{d^2_{i,j}\left(x,y\right)}\left(\frac{x-x^i_c}{d_{i,j}(x,y)}\vec{\boldsymbol{x}} + \frac{y-y^j_c}{d_{i,j}(x,y)}\vec{\boldsymbol{y}}\right), \tag{3}$$

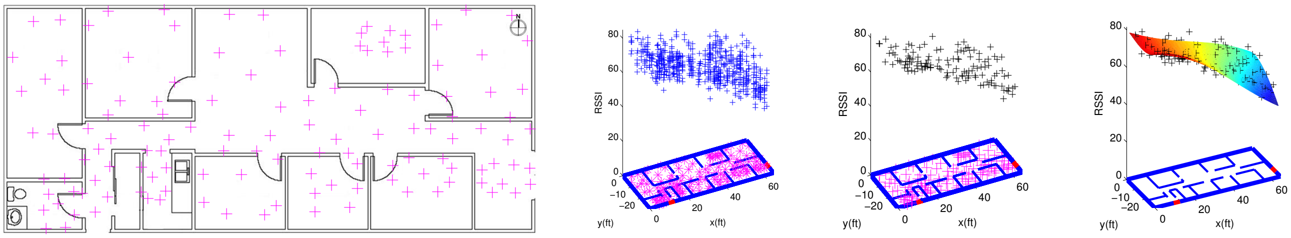
where $d_{i,j}(x,y)$ is the distance between the person's current position, (x,y), and the occupied cell position, (x^i_c, y^j_c). $\vec{\boldsymbol{x}}$ and $\vec{\boldsymbol{y}}$ are the unit vectors along the x and y direction. F_{cr} is the force constant and design parameter that controls the overall 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 results in trajectory estimates that pass through walls.

The total resultant force $\boldsymbol{F}_r(x,y)$ is the vectorial sum of forces exerted by all the occupied cells on the person's current cell location.

$$\boldsymbol{F}_r(x,y) = \sum_{i,j}\boldsymbol{F}_{i,j}\left(x,y\right). \tag{4}$$

This repelling force function $\boldsymbol{F}_r(x,y)$ is calculated once off-line, 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



(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.

yields the dynamic state-space model,

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

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

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

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

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 $\boldsymbol{v}_{p,k} = \begin{bmatrix} v_{p_{x,k}} & v_{p_{y,k}} \end{bmatrix}$ is modeled as zero mean white Gaussian.

2.2. Observation Model

As we have used 3 different sensor technologies for observations, the observation model shown in (2) depends on the specific technology used.

2.2.1. RSSI Observation model

A naive approach to using RSSI measurements involve 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 of RSSI tracking literature, the observation likelihood is approximated with a simple fixed *a priori* distribution (*e.g.*, Gaussian distribution) [4, 6]. 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. At each calibration point, a person carrying a body borne RSSI tag spends a fixed amount of time 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. Note the high variability of RSSI values at each location. The 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, a *RBF* network is used to fit a nonlinear map between known calibration locations and the mean RSSI observations as illustrated in Fig. 2(d). A *RBF* network is a feed forward neural network consisting of a hidden layer of radial kernels and an output layer of linear neurons [31]. A Gaussian kernel is used as the radial basis. This *RBF* map represents the forward generative observation model,

$$z_{m,k} = h_{m,k}\left(x_k, y_k\right) + n^r_{m,k} \quad (9)$$

where $z_{m,k}$ is the observed RSSI from access point $m, \;\; 1 \leq m \leq M$, with noise $n^r_{m,k}$ 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 m-th access point is specified by

$$h_{m,k}\left(x_k, y_k\right) = \boldsymbol{W}_m^T \boldsymbol{K}_{m,G}\left(\begin{bmatrix} x_k & y_k \end{bmatrix}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\right),$$

where $\boldsymbol{K}_{m,G}$ is the Gaussian kernel function with mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\boldsymbol{\Sigma}_m$ and can be defined as:

$$\boldsymbol{\mu}_m = \left[\begin{array}{cccc} \boldsymbol{\mu}_{m,1} & \boldsymbol{\mu}_{m,2} & \cdots & \boldsymbol{\mu}_{m,C} \end{array}\right]^T,$$

$$\boldsymbol{\Sigma}_m = \left[\begin{array}{cccc} \boldsymbol{\Sigma}_{m,1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_{m,2} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{\Sigma}_{m,C} \end{array}\right].$$

It is assumed that there are C Gaussian kernels in the hidden layer of the RBF network. $\boldsymbol{W}_m$ are the output layer linear weights,

$$\boldsymbol{W}_m = \left[\begin{array}{cccc} w_{m,0} & w_{m,1} & \ldots & w_{m,C-1} \end{array}\right].$$

The parameters of each Gaussian kernel $\{\boldsymbol{\mu}_{m,c}, \boldsymbol{\Sigma}_{m,c}\}$ and the hidden to output layer weights $\boldsymbol{W}_{m,c}$ are learned using a hybrid procedure that operates in 2 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 Expectation Maximization (*EM*) algorithm. The hidden to output weights are then calculated in a batch least square manner in order to minimize the MSE error at the output. Fig. 2(d) illustrates a nonlinear observation map learned from calibration data.

The observed RSSI $z_{m,k}$, RBF function $h_{m,k}$, and the observation noise $n^r_{m,k}$ from each access point are combined to form a multi-dimensional observation model,

$$\boldsymbol{z}_k = \left[\begin{array}{ccccccc} z_{1,k} & z_{2,k} & \ldots & z_{m,k} & \ldots & z_{M,k} \end{array}\right] \quad (10)$$

$$\boldsymbol{h}_k = \left[\begin{array}{ccccccc} h_{1,k} & h_{2,k} & \ldots & h_{m,k} & \ldots & h_{M,k} \end{array}\right] \quad (11)$$

$$\boldsymbol{n}^r_k = \left[\begin{array}{ccccccc} n^r_{1,k} & n^r_{2,k} & \ldots & n^r_{m,k} & \cdots & n^r_{M,k} \end{array}\right] \quad (12)$$

where $\boldsymbol{z}_k$ is the multi-dimensional RSSI observations emanating from M access points. Similarly $\boldsymbol{h}_k$ and $\boldsymbol{n}^r_k$ are the RBF observation model and the measurement noise for the M access points.

Once fit using calibration data, this RBF observation model may be used in the Bayesian framework for tracking. The model takes into account room specific multi-path and attenuation, and RSSI variability. By learning the map, the need to specify the location of the access points is also avoided.

2.2.2. *IR motion sensor Observation model*

Infra-red (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 mean value is taken to be a position inline with the orientation of the sensor at a distance based on approximate sensor range. The variance is based on the beam width of the sensor. Specific values for the mean and variance are found by characterizing the sensors and are somewhat arbitrary. The observation model is thus linear and defined as:

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}^{ms}_k \quad (13)$$

$$\boldsymbol{z}_k = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \boldsymbol{x}_k + \boldsymbol{n}^{ms}_k, \quad (14)$$

where $\boldsymbol{H}_k$ is the observation matrix and $\boldsymbol{n}^{ms}_k$ is the Gaussian observation noise with mean and variance as determined for the location and beamwidth of the sensor. 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.

2.2.3. *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,

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{n}^{f}_k \quad (15)$$

$$\boldsymbol{z}_k = \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right] \boldsymbol{x}_k + \boldsymbol{n}^{f}_k, \quad (16)$$

where $\boldsymbol{n}^f_k$ is the Gaussian observation noise for the foot-switch sensors. The mean value of the Gaussian is set to the known location of the switch. The variance specifies the accuracy of the localization for the foot switch.

2.2.4. *Multiple sensors observation model*

The Kalman framework allows for fusing multiple sensors of different types as available. An augmented observation vector is specified,

$$\boldsymbol{z}_k = \left[\begin{array}{ccc} \boldsymbol{z}^{RSSI}_k & \boldsymbol{z}^{IR}_k & \boldsymbol{z}^{Foot}_k \end{array}\right] \quad (17)$$

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.

3. SPKS BASED LOCATION TRACKER

The Kalman filter [32] provides the optimal Bayesian recursive estimate for the state $\boldsymbol{x}_k$ of a linear state-space system driven by Gaussian noise. 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. In contrast, the Kalman smoother estimates the conditional expectation of the state at time k given all past and future measurements $\boldsymbol{Z}_k = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_N]$, $1 \leq k \leq N$. Several common Kalman smoothing formulations are given in [19, 33–36]. In this article, we consider both fixed lag and fixed interval smoothing. In fixed lag smoothing, L future measurements are used at every time step to obtain smoothed state estimates at time k. In fixed-interval smoothing, the final time N is fixed and smoothed estimates are found using all N measurements.

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) [37], central difference Kalman filter (CDKF) [28], and their square-root variants [29], 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. This approach captures the posterior mean and covariance accurately to the 2nd-order (3rd-order is achieved for symmetric distribution) compared to the EKF which linearizes the nonlinear systems and only achieves 1st-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). Two SPKS variants are investigated, the forward-backward (FBSL-SPKS) [30] and fixed-lag (FL-SPKS) [33]. The FBSL-SPKS corresponds to a fixed interval smoothing approach whereby a SPKF is used to estimated 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 is found 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.

3.1. 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 linearizing the nonlinear model [38]. In this way, WSLR is more accurate in the statistical sense than first-order linearization, which does factor in 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 to generate smoothed estimates in the standard manner. Before presenting the pseudo-code for the SPKS, we first give a short review on the relationship between the SPKF and WSLR.

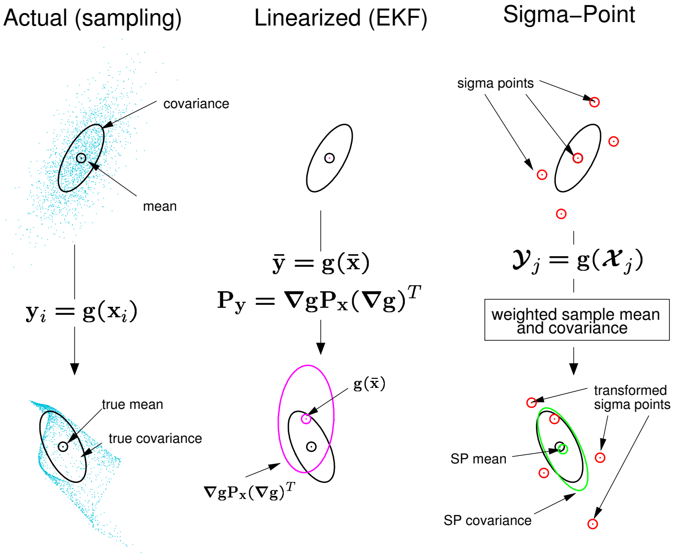

Actual (sampling)
Linearized (EKF)
Sigma−Point
covariance
mean
sigma points
$\mathbf{y}_i = \mathbf{g}(\mathbf{x}_i)$
$\bar{\mathbf{y}} = \mathbf{g}(\bar{\mathbf{x}})$
$\mathbf{P_y} = \nabla\mathbf{g}\mathbf{P_x}(\nabla\mathbf{g})^T$
$\mathcal{Y}_j = \mathbf{g}(\mathcal{X}_j)$
weighted sample mean and covariance
true mean
true covariance
$\mathbf{g}(\bar{\mathbf{x}})$
$\nabla\mathbf{g}\mathbf{P_x}(\nabla\mathbf{g})^T$
transformed sigma points
SP mean
SP covariance

Fig. 3. 2D 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.

Consider a prior random variable (RV) $\boldsymbol{x}$ which is propagated through a nonlinear function $g(\boldsymbol{x})$ to obtain a posterior RV $\boldsymbol{z}$. 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 $\boldsymbol{P_x}$

$$\boldsymbol{\chi} = \left[\begin{array}{ccc} \bar{\boldsymbol{x}} & \bar{\boldsymbol{x}} + \gamma\sqrt{\boldsymbol{P_x}} & \bar{\boldsymbol{x}} - \gamma\sqrt{\boldsymbol{P_x}} \end{array}\right] \tag{18}$$

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 $\boldsymbol{\gamma}_i$.

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

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 2D distribution. 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 $\boldsymbol{x}$. Defining $J = \mathrm{E}\left[\boldsymbol{\epsilon}^T \boldsymbol{W} \boldsymbol{\epsilon}\right]$ is the expected mean square error with sigma-point weighting matrix $\boldsymbol{W}$,

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

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 use 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), $\hat{\boldsymbol{P}}_{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{P}_{\boldsymbol{x}}\boldsymbol{A}^T + \boldsymbol{P}_{\boldsymbol{\epsilon}}$, 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 in the minimum mean square error (MMSE) sense. The approach explicitly takes into account the prior RV statistics.

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

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k\left(\boldsymbol{x}_k, \boldsymbol{v}_k\right) \tag{31}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k\left(\boldsymbol{x}_k, \boldsymbol{n}_k\right) \tag{32}$$

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 [37] for more details).

Alternatively, the SPKF may be derived from the WSLR from of the nonlinear state space,

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_{f,k}\boldsymbol{x}_k + \boldsymbol{b}_{f,k} + \boldsymbol{G}_{f,k}\boldsymbol{v}_k + \boldsymbol{G}_{f,k}\boldsymbol{\epsilon}_{f,k} \tag{33}$$

$$\boldsymbol{z}_k = \boldsymbol{A}_{h,k}\boldsymbol{x}_k + \boldsymbol{b}_{h,k} + \boldsymbol{n}_k + \boldsymbol{\epsilon}_{h,k} \tag{34}$$

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 error with mean zero and covariance $\boldsymbol{P}_{\boldsymbol{\epsilon}_f,k}$ and $\boldsymbol{P}_{\boldsymbol{\epsilon}_h,k}$. 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 [38]). The key advantage, however, of the statistically linearized form is that it can be used to derive the necessary backward filter. The FBSL-SPKS equations, consisting of the forward filter, backward filter, and smoother, are specified in the next sections.

3.1.1. 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 (31) and (32), are written below using the WSLR formulation:

- *Initialization*:

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

For k=1,2,…,N

- *Calculate sigma-points*:

$$\boldsymbol{\chi}_k^a = \left[\begin{array}{ccc} \hat{\boldsymbol{x}}_k^a & \hat{\boldsymbol{x}}_k^a + \Lambda & \hat{\boldsymbol{x}}_k^a - \Lambda \end{array}\right]$$

where $\Lambda = \sqrt{\left(L+\lambda\right)\boldsymbol{P}_{\boldsymbol{x}_k}^a}$

- *Time-update equations*:

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

- *Weighted Statistical Linearization of f(.)*:

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

- *Measurement-update equations*:

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

- *Weighted Statistical Linearization of h(.)*:

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

- *where*:

$$\begin{aligned}
\boldsymbol{x}^a &= \left[\begin{array}{ccc} \boldsymbol{x}^T & \boldsymbol{v}^T & \boldsymbol{n}^T \end{array}\right]^T \\
\boldsymbol{\chi}^a &= \left[\begin{array}{ccc} \left(\boldsymbol{\chi}^{\boldsymbol{x}}\right)^T & \left(\boldsymbol{\chi}^{\boldsymbol{v}}\right)^T & \left(\boldsymbol{\chi}^{\boldsymbol{n}}\right)^T \end{array}\right]^T \\
\boldsymbol{P}_{\boldsymbol{x}_k}^a &= \left[\begin{array}{ccc} \boldsymbol{P}_{\boldsymbol{x}_k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_k \end{array}\right]
\end{aligned}$$

- *Parameters*:
λ is the composite scaling parameter

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

and $w_i^{(c)}$ and $w_i^{(m)}$ are the scaler sigma-point weights defined as:

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

where α controls the size of the sigma point spread and should be within $0 \leq \alpha \leq 1$ to avoid sampling non-local points when the nonlinearities are strong [38]. $\beta \geq 0$ is the weighting term which incorporates the higher order moments of the prior distribution. The sigma point approach can effectively capture the first 2 moments (mean and covariance) of the distribution. The parameter β also can be used to minimize the error in approximating higher order moments of the distribution. For a Gaussian prior, we set $\beta = 2$ [37]. The parameter κ is used to ensure the positive definiteness of the covariance estimates. Setting $\kappa \geq 0$ should work for most cases. L is the dimension of the augmented state, $\boldsymbol{Q}_k$ 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.

3.1.2. 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 found during the forward pass (see [30]). The pseudo code is given below:

- Initializations:

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

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:

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

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

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

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

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

- Measurement-update equations:

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

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 in the time-update and measurement-update equations. This term is absent in the standard information filter formulation [19,33]. 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.

3.1.3. Smoothing

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

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

For tracking purposes, the FBSL-SPKS provides an off-line 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.

3.2. 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 augmenting state space,

$$\tilde{\boldsymbol{x}}_{k+1} = \begin{bmatrix} \boldsymbol{x}_{k+1} \\ \boldsymbol{x}_k \\ \vdots \\ \boldsymbol{x}_{k-L} \end{bmatrix} \tag{29}$$

$$= \begin{bmatrix} \boldsymbol{f}_k(\boldsymbol{x}_k) \\ \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}\tilde{\boldsymbol{x}}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{I} \\ 0 \\ \vdots \\ 0 \end{bmatrix}\boldsymbol{v}_k \tag{30}$$

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{n}_k \tag{31}$$

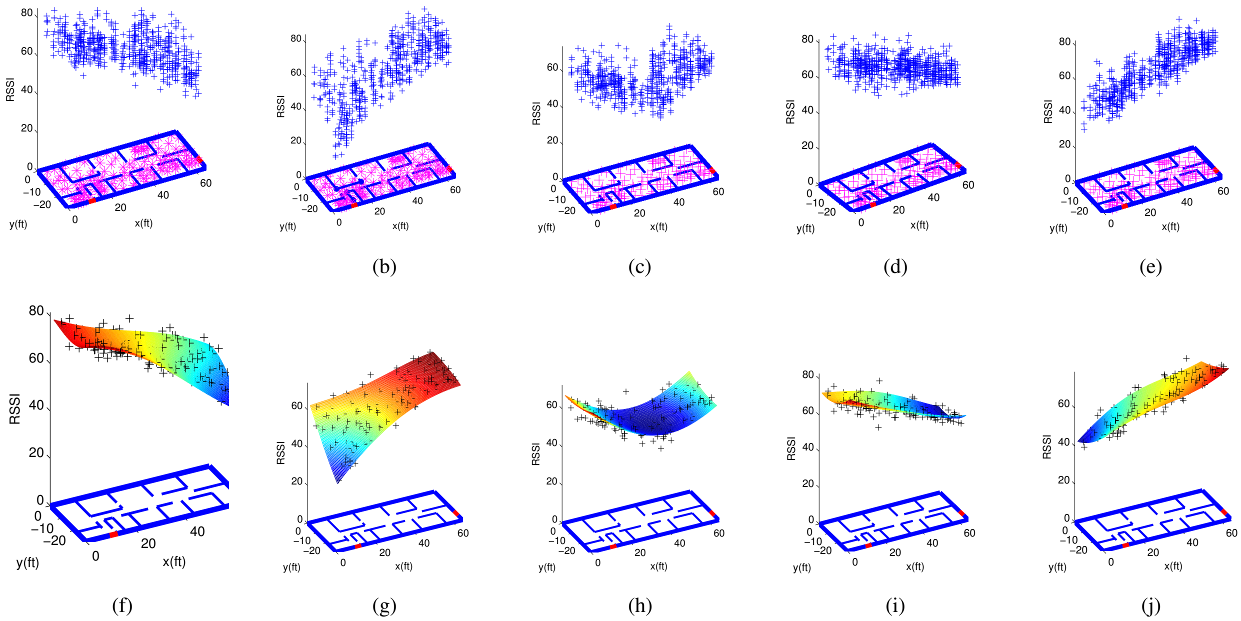
where

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

The standard SPKF recursions shown in 3.1.1 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 on-line implementation than the FBSL-SPKS.

4. EXPERIMENTAL RESULTS

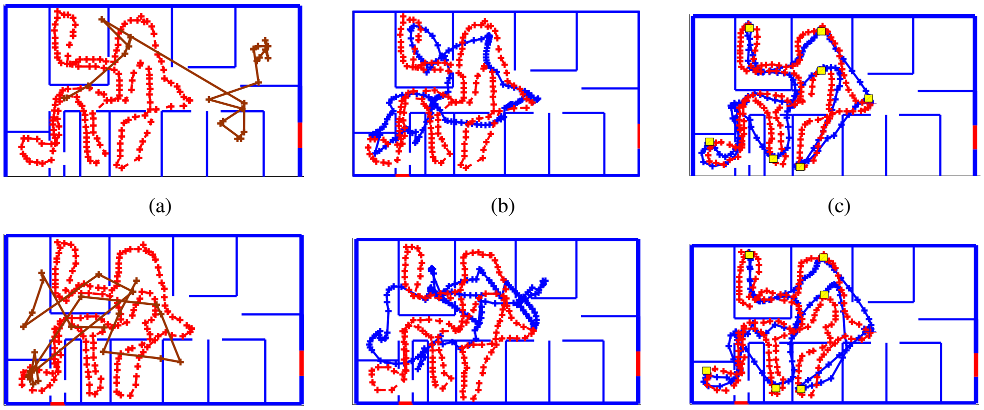
Implementation and testing was performed at several ”living-laboratories” (also called Point-of-Care labs) used to develop monitoring and assistive technologies for the elderly. A number of trials were conducted in which different subjects followed a predefined path. While walking, the subject periodically noted down the ground truth. The Ekahau real-time positional engine was also turned on during these tests for comparison. Note that the same calibration data was used for both the SPKS based tracker and Ekahau's positioning engine. The performances of SPKS based tracker and Ekahau real time tracking engine displayed in this paper were performed at 3 different sites.



(b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 4. (a) to (e): raw RSSI values from 5 access points collected during calibration at Point of Care test lab-I, (f) to (j): fitted RBF maps.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Tracking performance in test lab-I, (a) and (d) Ekahau estimates (red: ground truth, brown: estimate), (b) and (e) SPKS estimates using RSSI measurements (red: ground truth, blue: SPKS estimate), (c) and (f) SPKS estimates using RSSI + foot switch observations (red: ground truth, blue: SPKS estimate). Yellow rectangular boxes indicate the position of the foot switches on the floor plan. First row is for subject 1 and the second row is for subject 2.

4.1. Test Lab-I

The Point-of-care test lab-I was setup with 5 access points located at the four corners and at the center. The size of the test lab-I is 60 feet by 20 feet. 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 9 points were chosen to perform calibration in such a way that 8 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 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 5 access points were shown in Fig. 4(a)- 4(j).

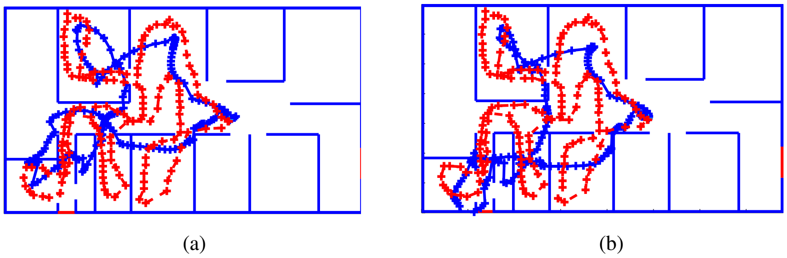

(a)
(b)

Fig. 6. (a) FBSL-SPKS estimates, (b) FL-SPKS estimates (red: ground truth, blue: SPKS estimates). All estimates used only RSSI sensors.

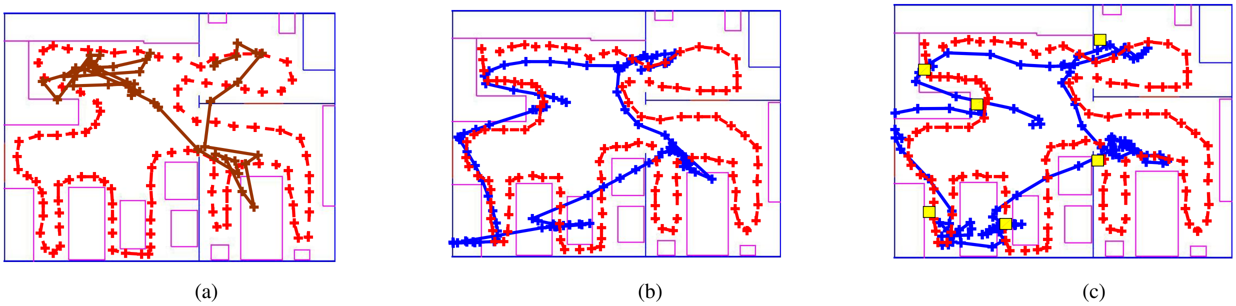

(a)
(b)
(c)

Fig. 7. Tracking performance in test lab-II, (a) Ekahau estimates (red: ground truth, brown: Ekahau estimate), (b) SPKS estimates using RSSI measurements (red: ground truth, blue: SPKS estimate), (c) SPKS estimates using RSSI + foot switch observations (red: ground truth, blue: SPKS estimate). Small yellow rectangular boxes indicate the location of the motion sensors on the floorplan. The furniture positions are shown as magenta rectangular boxes.

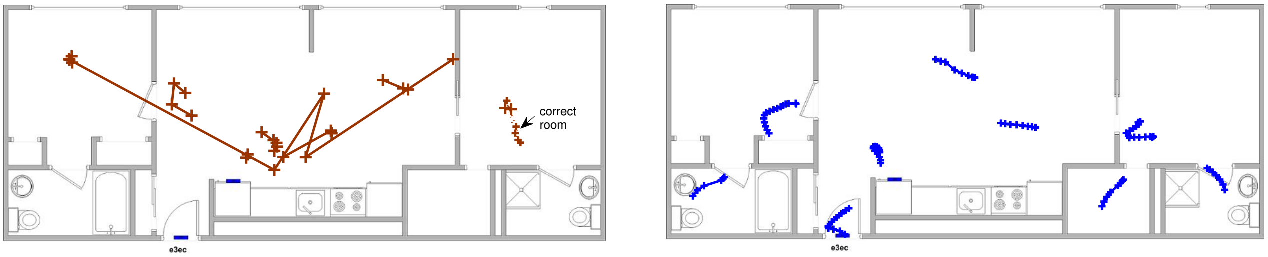
We conducted two trials of moving test in which subjects walked at a normal speed following a predefined path. Two different subjects were chosen as RSSI variability is observed to be subject dependent. Subject 1 took 174 seconds to complete the path and 25 RSSI observations were recorded during that time period. Subject 2 completed the same path in 169 seconds and recorded 21 RSSI observations. The rate varied between $4-8$ seconds during tracking. Multi-rate filtering was implemented so that the time-updates equations still provide estimates of the position and velocity every second. Approximate ground truth was collected periodically during the walking and is also shown in the plots. Fig. 5(a)- 5(f) compares the estimates obtained from the Ekahau engine and SPKS tracker. From Fig. 5(a) and 5(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 5(e)).

Only the FBSL-SPKS estimates are shown in the previous figures. The tracking performance of the FBSL-SPKS and FL-SPKS are compared in Fig. 6(a)- 6(b). As shown, the FL-SPKS estimates were 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 5(f)). Note that we set the variance of the foot-switch sensors to be 10 feet in our experiments. While this is clearly larger than necessary, the goal was to simulate an accuracy closer to that of the IR motion sensors rather than provide exact localization as would be possible with the foot-switches.

4.2. Test Lab-II

Similar to POCL test lab-I, calibration was performed in the other POCL test lab for each of the 5 Wi-Fi access points. The lab is also fitted with a number of IR motion sensors instead of foot switches. There are two varieties of motion sensor installed in the houses depending on the beam width. The full beam width unconstrained sensors generally installed one per room and has 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 7(a)- 7(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 with RSSI and RSSI with IR motion sensors are depicted in Fig. 7(b) and 7(c). While still superior to the Ekahau estimates, the small size of the POCL lab-II (30 feet by



(a)

(b)

Fig. 8. Tracking performance in test lab-III, (a) Ekahau estimates (brown: Ekahau estimate), (b) SPKS estimates using RSSI only measurements (blue: SPKS estimate).

20 feet) 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.

4.3. Test Lab-III

In 8(a)- 8(b), we demonstrate an additional moving test at a third location. The size of lab-III is 55 feet by 25 feet. Similar to test lab I and II, this living lab is also equipped with 5 wireless access points placed at four corners and at the center. The entire floorplan is divided into 9 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 the 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).

5. 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 incorporating 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.

Table 1. Performance comparison of the SPKS with other estimators. Average error is calculated relative to the observed true trajectory over 15 different trials.

Estimator	**Average Error** (ft)
Ekahau (RSSI)	12.22
EKF(RSSI)	6.58
EKS(RSSI)	4.80
SPKF(RSSI)	5.31
SPKS(RSSI)	3.45
SPKS(RSSI+IR motion sensor)	3.24
SPKS(RSSI+footswitch)	1.24

For the Bayesian inference, we use sigma point Kalman filters (SPKF), which provide improved performance over standard extended Kalman filters (EKF) while maintaining 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 1 summarizes the performance and superiority of the proposed SPKS based tracker over other popular estimation techniques in terms of average position error. The trials used in table 1 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. 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 environment or to even avoid the initial off-line 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.

6. ACKNOWLEDGMENTS

This work was supported in part by Intel as part of the Behavioral and Intervention Commons (BAIC). We would also like to thank Misha Pavel and Tamara Hayes for project support and use of laboratory facilities, and Ann Tsay, Eric Earl and Jon Yeargers for help with software support and data collection.

7. REFERENCES

[1] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 91–102, January 1992.

[2] P. Bahl and V. Padmanabhan, "Radar: An in-bulding rf-based user location and tracking system," in *Proceedings of IEEE Infocomm 00*, April 2000.

[3] N. Priyantha, A. Charraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of ACM Mobicomm 00*, July 2000.

[4] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitious computing: A case study," in *Proceedings of International Conference on Ubiquitious Computing (UBICOMP)*, 2004.

[5] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert5, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Proceedings of International Conference on Pervasive Computing (Pervasive)*, 2005.

[6] Y. Cheng, Y. Chawathe, A. LaMraca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *Proceedings of 3rd international conference on Mobile systems, applications, and services*, 2005, pp. 233–245.

[7] [Online]. Available: http://www.ekahau.com/

[8] [Online]. Available: http://www.ubisense.net/

[9] [Online]. Available: http://www.sonitor.com/

[10] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE pervasive computing*, 2003.

[11] A. Elfes, E. Prassler, and J. Scholz, "Tracking people in a railway station during rush hour," in *Proceedings of computer vision system, ICVS 99*, 1999.

[12] A. H. A. Fod and M. Mataric, "Laser-based people tracking," in *Proceedings of the IEEE International Conference on Robotics & Automation*, 2002.

[13] C.-T. Hsieh, C.-Y. Chen, and Y.-K. Wu, "People tracking system using kalman filter," in *Proceedings of Signal and image processing*, 2002.

[14] J. Krumm and E. Horvitz, "Locadio:inferring motion and location from wi-fi signal strengths," in *Proceedings of International Conference onMobile and Ubiquitious Systems:Networking and Services(MobiQuitous 04)*, 2004.

[15] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. Nordlund, "Particle filters for positioning, navigation and tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 425–435, 2002.

[16] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," *Proc. of Robotics: Science and Systems, 2006*, 2006.

[17] J. Letchner, D. Fox, and A. LaMarce, "Large-scale localization from wireless signal strength," *Proceeding of the National Conference on Artificial Intelligence (AAAI-05)*, 2005.

[18] D. Hhnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with rfid technology," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[19] F. Lewis, *Optimal Estimation with an Introduction to Stochastic Control Theory*. John Wiley and Sons, 1986.

[20] 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.

[21] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[22] S. Borthwick, M. Stevens, and H. Durrant-Whyte, "Position estimation and tracking using optical range data," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 1993.

[23] S. Seidal and T. Rappaport, “914 mhz path loss prediction models for indoor wireless communications in multifloored buildings,” *IEEE Transactions on Antennas and Propagation*, vol. 40(2), 1992.

[24] E. A. Wan and R. van der Merwe, “The unscented kalman filter for nonlinear estimation,” in *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, Octobor 2000.

[25] E. Wan and R. van der Merwe, *Kalman Filtering and Neural Networks*, 1st ed. John Wiley & Sons, 2001, ch. 7, pp. 221–280.

[26] R. van der Merwe and E. Wan, “Sigma-point kalman filters for probabilistic inference in dynamic state-space models,” in *Proceedings of Workshop on Advances in Machine Learning*, June 2003.

[27] S. Julier, J. Uhlmann, and H. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *Proceedings of American Control Conference*, 1995, pp. 1628–1632.

[28] M. Norgaard, N. Poulsen, and O. Ravn, “New developments in state estimation for nonlinear systems,” *Automatica*, vol. 36, pp. 1627–1638, November 2000.

[29] R. van der Merwe and E. A. Wan, “The square-root unscented kalman filter for state and parameter estimation,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6, Salt Lake City, Utah, May 2001, pp. 3461–3464.

[30] A. S. Paul and E. A. Wan, “A new formulation for nonlinear forward-backward smoothing,” in *Proceedings of International Conference on Acoustics, Specch and Signal Processing(ICASSP)*, 2008.

[31] S. Haykin, *Neural networks : a comprehensive foundation*. Maxwell Macmillan International, 1994.

[32] R. Kalman, “A new approach to linear filtering and prediction problems,” *Trnsactions of the ASME, Journal of Basic Engineering*, vol. 82, pp. 35–45, March 1960.

[33] D. Simon, *Optimal State Estimation*. Wiley Inter-Science, 2006.

[34] D. Fraser and J. Potter, “The optimum linear smoother as a combination of two optimum linear filters,” *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[35] H. Rauch, F. Tung, and C. Striebel, “Maximum likelihood estimates of linear dynamic systems,” *AIAA*, vol. 3, no. 8, pp. 1445–1450, 1965.

[36] C. Leondes, J. Peller, and E. Stear, “Nonliner smoothing theory,” *IEEE Transactions on System, Science and Cybernetics*, vol. 6, no. 1, January 1970.

[37] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, vol. 92, pp. 401–422, March 2004.

[38] R. van der Merwe, “Sigma-point kalman filters for probabilistic inference in dynamic state-space models,” Ph.D. dissertation, OGI School of Science and Engineering at Oregon Health & Science University (OHSU), 2004.