18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Particle Filtering for Positioning Based on Proximity Reports

Yuxin Zhao, Feng Yin, Fredrik Gunnarsson and Mehdi Amirijoo
Ericsson Research
Linköping, Sweden
Email: {first name.last name}@ericsson.com

Emre Özkan and Fredrik Gustafsson
ISY, Linköping University
Linköping, Sweden
Email: {emre, fredrik}@isy.liu.se

***Abstract*—The commercial interest in proximity services is increasing. Application examples include location-based information and advertisements, logistics, social networking, file sharing, etc. In this paper, we consider positioning of devices based on time series proximity reports from a mobile device to a network node. This corresponds to nonlinear measurements with respect to the device position in relation to the network nodes. Therefore, particle filtering is applicable for positioning. Positioning performance is evaluated in a typical office area with Bluetooth-low-energy beacons deployed for proximity detection and report. Accuracy is concluded to vary spatially over the office floor, and in relation to the beacon deployment density.**

I. Introduction

Particle filtering (PF) is known as a powerful numerical methodology for nonlinear and non-Gaussian Bayesian recursive estimation problems. The particle filtering theory has enabled many applications in various sectors over the past two decades [1]. One salient example is target tracking using radio signals, such as time-of-arrival (TOA), received-signal-strength (RSS), etc. Nowadays, it draws more attention to combine particle filtering/smoothing with cutting-edge techniques, such as cloud, big-data, beaconing, and sensor fusion techniques to meet the challenges in dealing with (1) sensor data complexity, (2) model complexity, and (3) large scale systems that are commonly encountered in practical problems—the main aim of the ongoing (2013-2017) European union training programme on Tracking in Complex Sensor Systems (TRAX) project [2].

With the deployment of small cells, Wi-Fi access points and Bluetooth-low-energy beacons, it is possible to utilize this infrastructure for positioning and thereby indoor location-based services. Motivated by this, we consider a novel RSS-proximity report based particle filtering for indoor positioning, where model complexity appears as a hard-thresholding when converting an RSS measurement into a proximity measurement. More precisely,

$$\text{Proximity} \triangleq \begin{cases} 0, & \text{RSS} \leq P_{th} \\ 1, & \text{RSS} > P_{th} \end{cases}. \tag{1}$$

A proximity measurement obtained in the above way reveals whether or not a target of interest is in the coverage area (depending on the threshold) of a *reference network node*, for instance a radio base station, a bluetooth-low-energy (BLE) beacon or a Wi-Fi access point. In a companion paper, we have also proposed a general framework for finding a reasonable RSS threshold, P_{th} [3]. Harness of proximity measurements may result in new fashioned positioning system with lower communication bandwidth, smaller database, as well as cheaper deployment and maintenance cost. Besides, there is a big trend nowadays in proximity based services, and such kind of quantized data can be used in many applications. To the best of our knowledge, RSS-proximity based PF for indoor positioning has never been considered before and serves as a working example of the theoretical work [4] on filtering and estimation using quantized sensor information.

Specifically, we consider centralized tracking of a mobile device using proximity measurements. The mobile device collects RSS measurements from the reference network nodes that know their own geographical positions and broadcast data package periodically to the air. In the conventional centralized framework, the mobile device will upload the RSS measurements to the *computation entity* either directly or via the reference nodes for position estimation. The computation entity is assumed to be equipped with sufficient computational power. In order to reduce the communication bandwidth and the storage usage at the computation entity, we propose to upload the proximity reports (converted from the RSS measurements according to (1)) to the computation entity.

Our contributions of this paper are as follows. First, we introduce an RSS-proximity model where the RSS threshold is selected in a more meaningful way as introduced in [3] and further develop a novel RSS-proximity report based particle filtering algorithm. Building map constraint is also taken into account to refine the position estimates. Then, evaluations are taken out with both simulated measurements as well as the real measurements collected within an office area. The resulting positioning system requires obviously less communication bandwidth and storage due to the binary nature of a proximity measurement.

The remainder of this paper is organized as follows: Section II introduces a complex state-space model and describes the problem at hand. Section III introduces a novel RSS-proximity based particle filtering algorithm. Section IV validates the new proposed RSS-proximity based particle filtering algorithm in various simulations. Lastly, Section V concludes the paper.

Throughout this paper, matrices are presented with uppercase letters and vectors with boldface lowercase letters. The operator $[\cdot]^T$ stands for vector/matrix transpose and $[\cdot]^{-1}$ stands for the inverse of a non-singular square matrix. The operator $\text{tr}(\cdot)$ denotes the trace of a square matrix. $\|\cdot\|$ stands for the Euclidean norm of a vector and $|\cdot|$ denotes the cardinality of a set. The operator $\mathbb{E}(\cdot)$ stands for the statistical expectation.

II. Models

We consider an indoor sensor network which comprises N_B reference network nodes with *a priori* known positions, $\mathbf{p}_j$, $j = 1, 2, \ldots, N_B$ and one computation entity. The state of a mobile device, $\boldsymbol{x}_k$, is to be tracked at each time instance k. To that end, the mobile device collects received-signal-strength measurements, converts them to proximity measurements, and further upload them to the computation entity for position filtering.

A. Propagation Models

This subsection provides a brief description of the linear log-distance propagation model. This model represents an RSS measurement in terms of the distance (one dimensional in nature) between a reference network node (say the jth) and the mobile device, namely,

$$r_j = \underbrace{A_j + 10B_j \log_{10}\left(\frac{d_j}{d_0}\right)}_{\mu_j} + e_j, \qquad (2)$$

where r_j is the RSS measurement, d_0 is the reference distance, A_j is the path loss measured at d_0, B_j is the path loss exponent, d_j is a short-hand notation of the Euclidean distance between a sample position $\mathbf{p}_\mathrm{r}$ and the jth reference network node's position $\mathbf{p}_j$, i.e., $d_j \triangleq ||\mathbf{p}_j - \mathbf{p}_\mathrm{r}||$. Often, the measurement noise e_j is assumed to be time invariant and univariate Gaussian distributed with zero mean and variance σ_j^2, i.e., $r_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$. The propagation model parameters, A_j, B_j, and σ_j^2, can be calibrated for instance in an offline phase, given a batch of RSS measurements collected from all reference nodes [3].

B. State-Space Model

In what follows, we merely consider state-space models that are linear in the state dynamics and non-linear in the measurements. For instance, we could use the following conventional motion model to relate position and velocity indoors [5],

$$\boldsymbol{x}_{k+1} = F\boldsymbol{x}_k + B_w \boldsymbol{w}_k,$$

where $\boldsymbol{x}_k = [p_{x_k}, p_{y_k}, v_{x_k}, v_{y_k}]^T$ is the underlying state vector at time instance k with p_{x_k} and p_{y_k} denoting the 2-D position and with v_{x_k} and v_{y_k} denoting the velocity in the corresponding dimension. $\boldsymbol{w}_k$ is the acceleration noise assumed to be multivariate Gaussian distributed with zero mean vector and covariance matrix $\Sigma_{\boldsymbol{w}} = \sigma_w^2 \mathbf{I}_2$. Two different measurement models, namely the conventional RSS model and the novel proximity model, are considered in the sequel. We start with the conventional RSS model, where the relationship between an RSS measurement and the state vector, say at time instance k, is given as follows:

$$\boldsymbol{r}_k = h(\boldsymbol{x}_k) + \boldsymbol{e}_k,$$

where $\boldsymbol{r}_k = [r_{k,1}, \ldots, r_{k,N_B}]^T$ and $h(\boldsymbol{x}_k) \in \mathbb{R}^{\dim(\boldsymbol{x}_k) \mapsto \dim(\boldsymbol{r}_k)}$. Considering the propagation model given in Section II-A and the assumption that RSS measurements collected for different reference nodes are mutually independent yields

$$\boldsymbol{r}_k = \begin{bmatrix} r_{k,1} \\ r_{k,2} \\ \vdots \\ r_{k,N_B} \end{bmatrix} = \begin{bmatrix} \mu_{1,k} + e_1 \\ \mu_{2,k} + e_2 \\ \vdots \\ \mu_{N_B,k} + e_{N_B} \end{bmatrix}$$

where $\mu_{j,k} \triangleq A_j + 10B_j \log_{10}(d_{k,j})$, and $d_{k,j}$ is the Euclidean distance between a sample position, $[p_{x_k}, p_{y_k}]^T$, and the jth reference node's position, $\mathbf{p}_j$, e_j is the measurement error defined in (2), which is Gaussian distributed with zero mean and variance σ_j^2.

For proximity based measurements, the proximity reports vector can be obtained by comparing the RSS measurements with a predefined threshold, as was shown in (1). A proximity measurement model is thus expressed as

$$\boldsymbol{y}_k = f(\boldsymbol{r}_k) = \begin{bmatrix} f(r_{k,1}) \\ f(r_{k,2}) \\ \vdots \\ f(r_{k,N_B}) \end{bmatrix} \qquad (3)$$

where $\boldsymbol{y}_k = [y_{k,1}, \ldots, y_{k,N_B}]^T$ is the proximity report vector with '1 'and '0 'indicating in and out of proximity, respectively. $f(x)$ is a non-linear function which performs hard-thresholding of an input, x, as follows:

$$f(x) = \begin{cases} 0, & x \leq P_{th} \\ 1, & x > P_{th} \end{cases}.$$

It is noted that in the RSS measurement model, the noise factor is additive and thus linear to the measurement vector. However, in the proximity model, nonlinear noise factor is introduced by the nonlinearity of function $f(x)$. Solving filtering problems with this nonlinearity will be the main focus of the remainder of this paper.

III. Particle Filtering Algorithm

Considering systems that are described by the state-space model in II-B, particle filtering algorithms based on both RSS and proximity measurements are given in this section. Particle filters are methods to perform Monte Carlo approximations to the optimal Bayesian filtering equations. The main problem with Bayesian filtering is to compute the posterior probability density $p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k})$, where $\boldsymbol{z}_{0:k}$ are all measurements collected since $k = 0$. Then, the minimum mean square (MMS) estimate is obtained by

$$\mathbb{E}[\boldsymbol{x}_k|\boldsymbol{z}_{0:k}] = \int \boldsymbol{x}_k p(\boldsymbol{x}_k|\boldsymbol{z}_{0:k})\, \mathrm{d}\boldsymbol{x}_k \qquad (4)$$

However, such an integral can be evaluated in closed form only in a few special cases (e.g., if the noise distribution of $\boldsymbol{e}_k$ and $\boldsymbol{w}_k$ are independent and Gaussian distributed, the measurement are linear in the state, i.e., $\boldsymbol{z}_k = C\boldsymbol{x}_k$, the optimal solution is given by the Kalman filter [6]). For more general cases, numerical methods have to be used. Monte Carlo methods provide numerical methods for calculating integrals of the form (4) by drawing samples from the distribution and estimating the quantities by sample averages. For the state-space model as given in Section II-B, we use the sequential importance resampling (SIR) method to approximate the posterior distribution. Further details of Monte Carlo approximation, Bayesian filtering and sequential importance sampling can be found in [6]. In the following subsections, we first give the choices of different importance distribution and then particle filter algorithms based on both RSS measurements and proximity reports.

A. Discussions on the Choice of Importance Distribution [7]

In practical Bayesian models, it is not possible to obtain samples directly from the posterior distribution due to its complicated functional form. In importance sampling (IS), we use an approximate distribution called the importance distribution, from which we can easily draw samples. Note that in following algorithms, the importance distribution from which the particle samples are selected is modeled as $p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}^{(i)})$. The choice of importance distribution may influence the depletion problem significantly. Here some comments on available options of the importance distribution are provided. The most general importance distribution has the form $\pi(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{0:k})$, where $\boldsymbol{z}_{0:k}$ denotes the measurements in general from time 0 to k and $\boldsymbol{x}_{0:k}$ denotes the state vector from time 0 to k. This means that the whole trajectory should be sampled at each iteration, which is not desirable in real-time applications. The general form can be factorized as

$$\pi(\boldsymbol{x}_{0:k}|\boldsymbol{z}_{0:k}) = \pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{0:k})\pi(\boldsymbol{x}_{0:k-1}|\boldsymbol{z}_{0:k})$$

The most common approximation is to reuse the path $\boldsymbol{x}_{0:k-1}$ and only sample the new state $\boldsymbol{x}_k$. Due to the Markov property assumed on the model, we have

$$\pi(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}, \boldsymbol{z}_{0:k}) = \pi(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_k)$$

The optimal distribution that includes all information from the previous state and the current observation can be derived as

$$\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k) = p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k) = \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)} \mid \boldsymbol{x}_{k-1}^{(i)})}{p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1}^{(i)})}$$

The weights are derived to be

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{z}_k)} \propto w_{k-1}^{(i)} p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1}^{(i)}).$$

However, it is generally hard to sample from this distribution and also the weight update, since it requires integrating over the whole state space

$$p(\boldsymbol{z}_k|\boldsymbol{x}_{k-1}^{(i)}) = \int p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})\mathrm{d}\,\boldsymbol{x}_k^{(i)}. \quad (5)$$

As an alternate, the likelihood function $p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})$ is often used as the importance distribution. However, in our work due to the form of the likelihood function as shown in (7), drawing samples from it is quite challenging. Hence, in this paper, the conditional prior of the state vector $p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(i)})$ is used as the importance distribution for simplicity but at the cost of losing useful information in $\boldsymbol{z}_k$. Efficient sampling from more advanced importance distributions will be our future work.

B. Particle Filtering Based on RSS measurements

According to the state-space model given in Section II-B, a particle filter algorithm based on RSS measurements is given below:

1) *Initialization*: Generate samples $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$, $i = 1, ..., N$. Here, $p(\boldsymbol{x}_0)$ denotes the prior probability of the initial state. Each sample of the state vector is referred to as a *particle*. Set $w_0^{(i)} = 1/N$, for all $i = 1, ..., N$.
2) *Importance Sampling*: For each $k = 1, ..., T$, do the following.
   a) Draw samples $\boldsymbol{x}_k^{(i)}$, $i = 1, ..., N$ from the importance distribution $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$.
   b) Calculate new weights according to

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\boldsymbol{r}_k|\boldsymbol{x}_k^{(i)})$$

   where $\boldsymbol{r}_k$ is a vector of RSS measurements and $p(\boldsymbol{r}_k|\boldsymbol{x}_k^{(i)})$ is the likelihood function of $\boldsymbol{r}_k$ given $\boldsymbol{x}_k^{(i)}$, which is given by:

$$p(\boldsymbol{r}_k|\boldsymbol{x}_k^{(i)}) = \prod_{j=1}^{N_B} p(r_{k,j}|\boldsymbol{x}_k^{(i)}) \quad (6)$$

   c) Normalize the weights to sum to unity, i.e.,

$$w_k^{(i)} := \frac{w_k^{(i)}}{\sum_{i=1}^N w_k^{(i)}}.$$

   d) The approximation to the posterior expectation of $\boldsymbol{x}_k$ is then given as

$$\hat{\boldsymbol{x}}_k \approx \sum_{i=1}^N w_k^{(i)} \boldsymbol{x}_k^{(i)}$$

3) *Resampling*: In resampling if the effective number of particles,

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2},$$

   is too low, perform resampling as follows:
   a) Interpret each weight $w_k^{(i)}$ as the probability of obtaining the sample index i in the set $x_k^{(i)} : i = 1, ..., N$.
   b) Draw N samples from that discrete distribution and replace the old sample set with this new one.
   c) Set all weights to the constant value $w_k^{(i)} = 1/N$.

The purpose of resampling is to prevent high concentration of probability mass at a few particles. Without this step, some $w_k^{(i)}$ will converge to 1 and the filter would brake down to a pure simulation.

Remark: In (6), we assume that the RSS measurements collected for different reference nodes are independent. Moreover,

$$p(r_{k,j}|\boldsymbol{x}_k^{(i)}) = p(\underbrace{A_j + 10B_j \log_{10}(d_{k,j}^{(i)})}_{\mu_{j,k}^{(i)}} + e_j|\boldsymbol{x}_k^{(i)})$$

$$= \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(r_{k,j} - \mu_{j,k}^{(i)})^2}{2\sigma_j^2}\right]$$

where $d_{k,j}^{(i)}$ is the Euclidean distance between the ith sample position, $\boldsymbol{x}_k^{(i)}$, and the jth reference node's position.

C. Particle Filtering Based on Proximity Reports

For proximity measurements, we follow the same particle filtering procedure as described above. However, the complexity of calculating marginal probability $p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})$ is increased by introducing a nonlinear noise factor. It further impacts the calculation of the weights. Accordingly, the method of calculating weights in step $2.b$ is given in the following equation:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})$$

where $\boldsymbol{y}_k$ is a vector of proximity measurements, cf.(3), and $p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)})$ is the probability distribution of $\boldsymbol{y}_k$ given $\boldsymbol{x}_k^{(i)}$, which is given by:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) = \prod_{j=1}^{N_B} p(y_{k,j}|\boldsymbol{x}_k^{(i)})$$

$$= \prod_{j=1}^{N_B} \sum_{l \in \{0,1\}} \delta(y_{k,j} - l) p(y_{k,j} = l|\boldsymbol{x}_k^{(i)}). \quad (7)$$

It is easy to prove further that

$$p(y_{k,j} = 0|\boldsymbol{x}_k^{(i)}) = \Phi\left(\frac{P_{th} - \mu_{k,j}^{(i)}}{\sigma_j}\right),$$

$$p(y_{k,j} = 1|\boldsymbol{x}_k^{(i)}) = 1 - p(y_{k,j} = 0|\boldsymbol{x}_k^{(i)}),$$

where

$$\Phi\left(\frac{t-\mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{t} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right].$$

IV. Results

In this section, descriptions of the evaluation setup as well as positioning performance of the proposed filtering algorithm will be provided.

A. Setup

We consider a typical office environment at Ericsson, Linköping, Sweden. In total $N_B = 10$ BLE beacons are placed rather uniformly in the area. The floor plan as well as the known beacon positions are shown in two-dimensional (2-D) space in Fig. 1. Herein, a local coordinate system is used. The BLE beacons serve as transmitters and broadcast beacon information regularly. The transmit power is $P_T = -58$ dBm. A moderate scale measurement campaign was conducted during normal work hours. Throughout the measurement campaign, the mobile device (equipped with BLE chipset) receives data packages from the BLE beacons and measures the RSS. In order to train a set of propagation parameters, a total number of $M = 12144$ RSS measurements were collected along 52 predefined tracks. After the training phase, an RSS propagation model between the transmitter and receiver is obtained as shown in Fig. 2. Then, propagation parameters are used to calculate the optimal RSS threshold. A brief description of the procedure to select a threshold is as follows: given a set of candidate RSS thresholds, evaluate the average positioning accuracy metric at each candidate threshold. Obtain the optimal threshold as the one optimizing an adequate positioning accuracy metric. One example of the impacts of selecting different thresholds on positioning accuracy will be given in later sections. For further details about the setup, training phase experiments, and selection of the threshold, consult to [3].

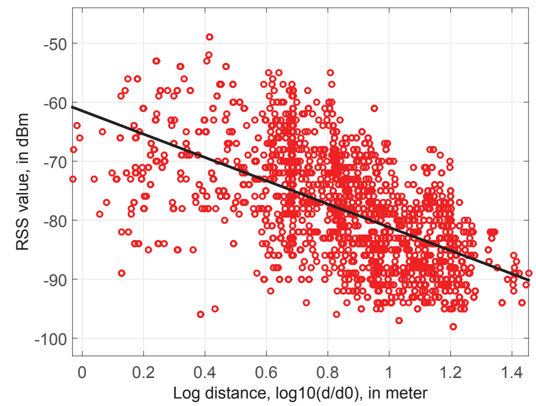

Fig. 2. Scatter plot of the collected RSS measurements (marked by red circles) versus the calibrated log-distance model (black line) for the 4th BLE beacon. The calibrated parameters for the log-distance model are $\hat{A}_4 = -60.0145$ dB, $\hat{B}_4 = -2.1156$ dB, and $\hat{\sigma}_4 = 7.45$ dB

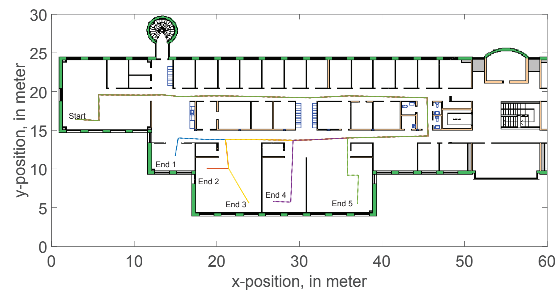

Fig. 3. Example of 5 pre-generated trajectories.

After the propagation parameters as well as the threshold are obtained, the evaluation of the proposed particle filter algorithms are performed. Two cases are evaluated in this section. In the first case, the measurements are simulated based on the propagation model given in II along the selected trajectories. The trajectories are either selected from one of the 52 tracks which are pre-defined to collect RSS measurements or generated according to the map constraints of the office area. Some examples of the generated trajectories are illustrated in Fig. 3. In order to have enough time for the particle filter to converge, usually the start and end points are selected such that each trajectory has at least 300 time stamps. In the second case,

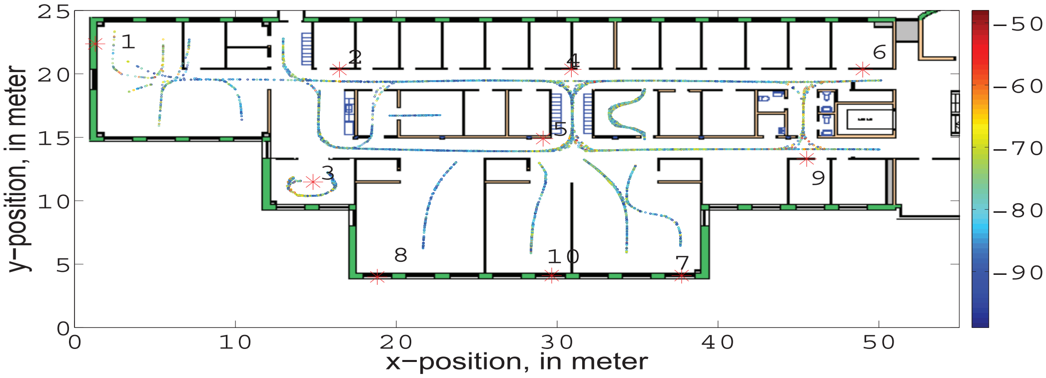

Fig. 1. Illustration of sensor deployment and the calibration set of sample positions and RSS measurements(with the strength in dBm indicated by different colors). The BLE beacons are indexed and marked by red $*$.

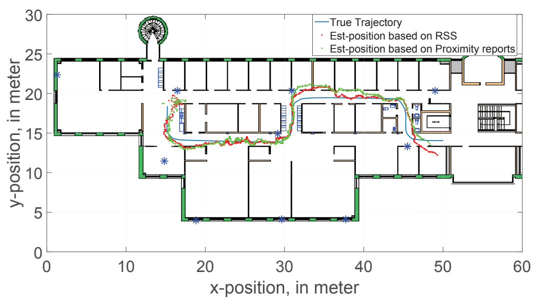

Fig. 4. Illustration of estimated positions and true trajectory with $\sigma_w = 0.5$.

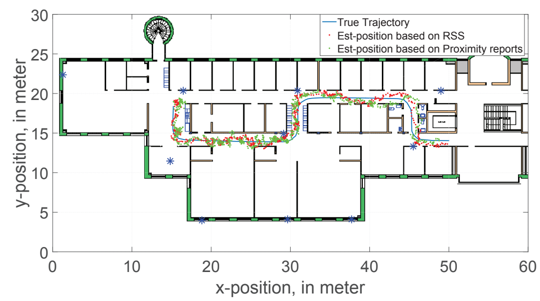

Fig. 6. Illustration of estimated positions and true trajectory with $\sigma_w = 2.5$.

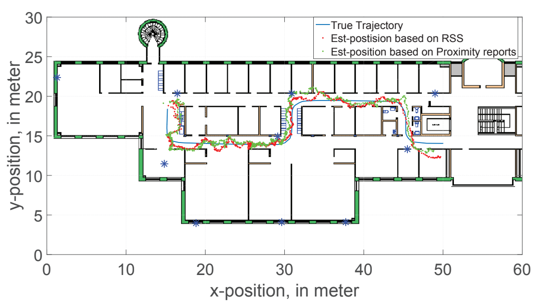

Fig. 5. Illustration of estimated positions and true trajectory with $\sigma_w = 1$.

TABLE I. EVALUATION PARAMETERS.

Parameter	Value	Description
F	$\begin{bmatrix}1 & 0 & T_s & 0\\ 0 & 1 & 0 & T_s\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$	CV motion model parameter
B_w	$\begin{bmatrix}\frac{T_s}{2} & 0\\ 0 & \frac{T_s}{2}\\ T_s & 0\\ 0 & T_s\end{bmatrix}$	CV motion model parameter
T_s	0.1 sec	Sampling interval
σ_w	1 or specified	Variance of the acceleration noise
P_{th}	-82 dBm	Predefined RSS threshold
N	5000 or specified	Number of particles
N_B	10	Number of deployed reference nodes
μ_0	$\boldsymbol{x}_0^*$	Mean of $\boldsymbol{x}_0$
Σ_0	$\begin{bmatrix}20 & 0 & 0 & 0\\ 0 & 20 & 0 & 0\\ 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 2\end{bmatrix}$	Covariance matrix of $\boldsymbol{x}_0$
M	20	Number of Monte Carlo runs
N_h	$\frac{2}{3}N$	Threshold for resampling

in order to show the performance of the proposed algorithm under real circumstances, we also provide results with real RSS measurements which are collected along the 52 predefined tracks. The RSS values are then converted to proximity measurements $\boldsymbol{y}_k$ by comparing RSS with the threshold. Then, the Constant Velocity (CV) motion model listed in Table I, is used to model the trajectories, since it matches well with human walking in indoor environment. Performance of particle filtering algorithms with various parameters listed in Table-I will be compared in Section IV-B.

B. Performance Evaluation

In this section, the positioning performance of the proposed algorithm is evaluated. In the following parts, the positioning performance with simulated measurements is given first. Then, the positioning performance for the case with real measurements collected from the office area is provided.

1) Performance evaluation with simulated measurements: In order to provide an overall performance of the filtering algorithm, an illustration of the estimated positions as well as the true trajectory are shown in Figure 4 to Figure 6 with different values of σ_w. This trajectory is selected from one of the 52 predefined tracks. It starts from the left side of the map (i.e. coffee room) to the right side (i.e. corridor next to the washroom) and consists of a number of 90 degrees turns. Particle filters are ran for several times (e.g. 10 times) and the position estimates averaged over all these Monte Carlo runs are plotted on the figures. It can be seen that algorithm based on proximity reports exhibits similar performance as compared to algorithm based on RSS measurements. The first a few estimated positions are a bit out of the track. This is probably due to the fact that in this area, there are fewer number of beacons which are within the communication range. Also, at the end of the trajectory, the positioning performance is degrading as it goes out of the coverage of all the beacons. In order to compare the statistical performance of the proposed filtering algorithm, the particle filter is applied for different trajectories and Monte Carlo evaluations are performed. Figure 3 shows 5 trajectories generated with the map constraints. All these 5 trajectories start at the same position and end at 5 different positions. Then, particle filtering is applied to each of these trajectories for M times. The estimation error at each position is calculated as

$$E_k = \sqrt{(\hat{p}_{x_k} - p^*_{x_k})^2 + (\hat{p}_{y_k} - p^*_{y_k})^2}$$

where $\hat{p}_{x_k}$, $\hat{p}_{y_k}$ denote the estimated position and $p^*_{x_k}$, $p^*_{y_k}$ denote the ground truth. The CDF of the estimation errors at each position is illustrated in Figure 7. Different error distribution has been observed for various σ_w values. Similar performance is achieved with $\sigma_w = 1$ and $\sigma_w = 2.5$, both of which out perform $\sigma_w = 0.5$. The result obtained here is not surprising , since larger σ_w is preferred to use when maneuvers dominate. However, larger σ_w will lead to larger gate, which is not desirable when dealing with multiple-target tracking. Hence, $\sigma_w = 1$ will be used in the following evaluations.

To compare the overall positioning accuracy with various parameters, the average RMSE is introduced as

$$\text{RMSE} = \sqrt{\frac{1}{N_{tr}} \sum_{tr=1}^{N_{tr}} (\frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{T_{tr}} \frac{1}{T_{tr}} E^2_{k,m,tr})}$$

where T_{tr} is the length of each trajectory and N_{tr} is the number of trajectories. The average RMSE versus the number of particles is shown in Figure 8. Proximity based particle filtering algorithm provides better accuracy when the number of particles is small (e.g. below 1000). Overall, approximately 0.25 m positioning accuracy difference can be seen between RSS and proximity based filtering for large number of particles. It can be concluded that proximity based algorithm is preferred for low computation complexity, and both RSS and proximity based algorithm can provide satisfactory accuracy when computation complexity is not essential. As discussed in [3], the selection of RSS threshold is critical in determining positioning accuracy. The comparison between various RSS thresholds is shown in Figure 9. Among all the tested P_{th} from -91 dBm to -70 dBm, the lowest RMSE is achieved at $P_{th} = -82$ dBm, which is consistent with the result in [3], although different criteria have been selected. This is reasonable since with a high threshold (e.g. -70 dBm), the receiver may not within any beacon's coverage, while a low threshold may lead to the case that receiver is within all beacon's coverage. In both cases, little information can be obtained, and the positioning accuracy may degrade due to the loss of information.

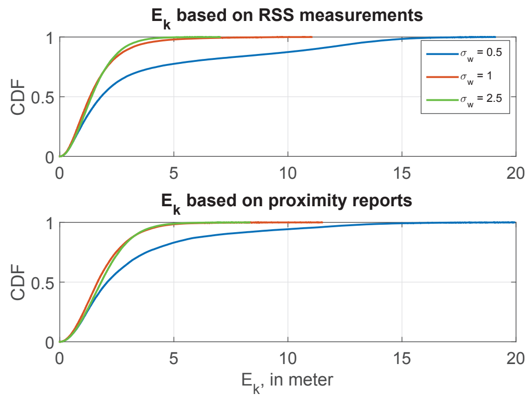

Fig. 7. Particle filter based on RSS measurements and proximity reports: CDF of E_k with different values of σ_w.

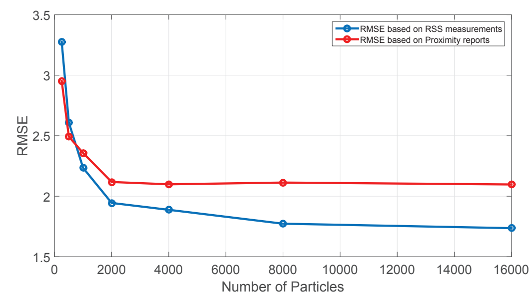

Fig. 8. RMSE versus number of particles

2) Performance evaluation with real measurements: In the above theoretical demonstration, it is assumed that RSS measurements from all beacons in the deployment area can be obtained. However, in practical environments, this is not realistic since the device may only measure RSS from 1 or 2 beacons at one position due to different scanning time of each beacon. In the following part, in order to show the performance of the proposed algorithm in practical circumstances, evaluations based on real RSS measurements taken out in the office area are provided. Figure 10 shows an overall performance with real RSS measurements which are collected within the office area. Similar performance is still achieved for proximity based positioning compared with RSS based positioning. The median estimation error is approximately 4 meters as shown in Figure 11. The filtering algorithm works well in distinguishing if the device is in the upper or lower corridor area. It is noted from the figures that in real circumstances, due to the complex indoor propagation conditions, the estimation of positions may be difficult. E.g. near the lower corridor on the figure, the user may also hear strong signals from beacon 8 and 10, so that the user may have been "fooled" that it is close to beacon 8 and 10. This may explain why the estimated positions for those points are very close to beacon 8 and 10. Considering all those factors which may impact the RSS propagation model, such as concrete walls penetration, reflections, multi-paths, shadowing effects, etc, the simple propagation model used here may not be sufficient to achieve similar performance as the case with

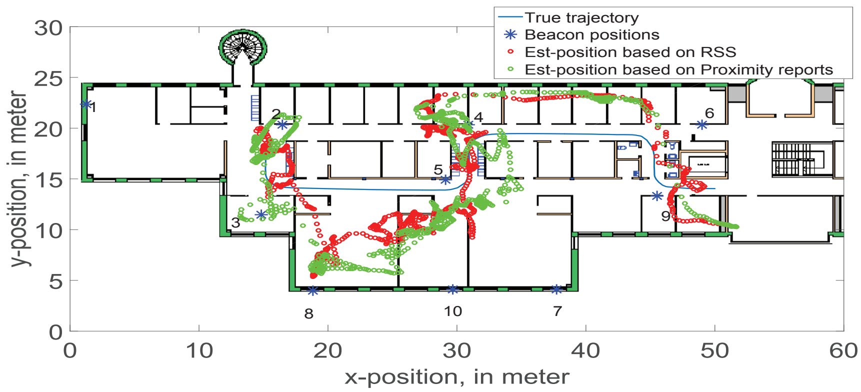

Fig. 10. Illustration of estimated positions based on real measurements.

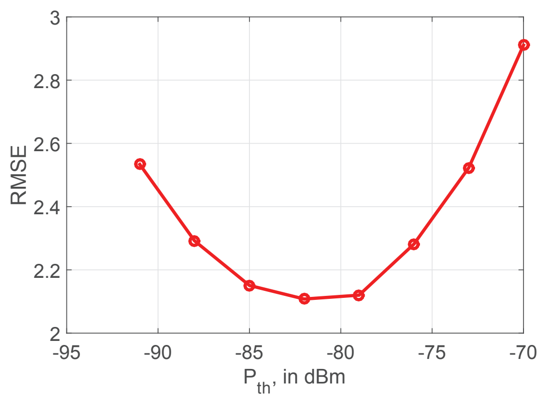

Fig. 9. RMSE versus different P_{th}

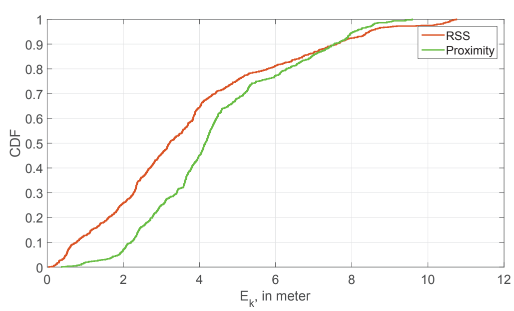

Fig. 11. CDF of positioning error based on real measurements.

simulated measurements. More advanced modeling may be studied in the future work. Moreover, due to the fact that the device may only receive RSS measurements from 1 or 2 beacons at a certain location, with only RSS measurements it is not sufficient for high accuracy positioning. However, the performance with the proposed algorithm is still acceptable for some applications which do not require high positioning accuracy.

V. Conclusions

In this paper, we have proposed a particle filtering algorithm which is applicable for indoor positioning based on time series proximity reports from a mobile device to the reference network node. Positioning performance of the proposed algorithm is comparable with that of the conventional one using RSS measurements. We have further demonstrated that the selection of a proper motion model, the particle numbers, the RSS threshold which determine proximity and the spacial distribution of beacons are all essential in determining positioning accuracy. Such a proximity based positioning algorithm is beneficial in different aspects, among which the signaling overhead can be significantly reduced via sending binary proximity reports (i.e. 1 bit) in stead of e.g., 8 bits (signed) quantized RSSI values.

Acknowledgment

This work is funded by the European Union FP7 Marie Curie training programme on Tracking in Complex Sensor Systems (TRAX) with grant number 607400. Furthermore, we acknowledge the support from SenionLab, who provided the BLE beacons as well as associated positioned RSS measurement data.

References

[1] F. Gustafsson, *Statistical Sensor Fusion*, 2nd ed. Lund, Sweden: Studentlitteratur, 2012.

[2] "European union fp7 tracking in complex sensor systems (trax) project," https://www.trax.utwente.nl/.

[3] F. Yin, Y. Zhao, and F. Gunnarsson, "Proximity report triggering threshold optimization for network-based indoor positioning," in *Submitted to Proc. Int. Conf. on Information Fusion*, 2015.

[4] R. Karlsson and F. Gustafsson, "Filtering and estimation for quantized sensor information," in *13th European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, Sept. 2005.

[5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[6] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press., 2013.

[7] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.