# Sequential Monte Carlo Methods and Theoretical Bounds for Proximity Report Based Indoor Positioning

Yuxin Zhao, Carsten Fritsche, Feng Yin, Fredrik Gunnarsson and Fredrik Gustafsson

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-147834

N.B.: When citing this work, cite the original publication.

Zhao, Y., Fritsche, C., Yin, F., Gunnarsson, F., Gustafsson, F., (2018), Sequential Monte Carlo Methods and Theoretical Bounds for Proximity Report Based Indoor Positioning, *IEEE Transactions on Vehicular Technology*, 67(6), 5372-5386. https://doi.org/10.1109/TVT.2018.2799174

Original publication available at:

https://doi.org/10.1109/TVT.2018.2799174

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

http://www.ieee.org/index.html

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.





Sequential Monte Carlo Methods and Theoretical Bounds for Proximity Report Based Indoor Positioning

Yuxin Zhao, Student Member, IEEE, Carsten Fritsche, Member, IEEE, Feng Yin, Member, IEEE, Fredrik Gunnarsson, Senior Member, IEEE, Fredrik Gustafsson, Fellow, IEEE

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 a time series of 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. Motion model will be needed together with the measurements to determine the position of the device. Therefore, sequential Monte Carlo methods, namely particle filtering and smoothing, are applicable for positioning. Positioning performance is evaluated in a typical office area with Bluetooth-low-energy beacons deployed for proximity detection and report, and is further compared to parametric Cramér-Rao lower bounds. Finally, the position accuracy is also evaluated with real experimental data.

Index Terms—Proximity, indoor positioning, particle filtering and smoothing, Cramér-Rao lower bounds.

#### I. Introduction

## A. Background

Particle filtering (PF) and particle smoothing (PS) are known as powerful numerical methods for nonlinear and non-Gaussian Bayesian recursive estimation problems. The particle filtering and smoothing theory has enabled many applications in various sectors over the past two decades [2]. 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 [3].

For indoor positioning, there are many techniques proposed in literature. For instance, a complete discussion on indoor

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org. This work is an extension of our conference paper [1].

Y. Zhao and F. Gunnarsson are with Ericsson Research, Linköping, SE-58330, Sweden. (E-mail: firstname.lastname@ericsson.com).

C. Fritsche and F. Gustafsson are with the Department of Electrical Engineering, Division of Automatic Control, Linköping University, Linköping, SE-58183, Sweden. (E-mail: carsten, fredrik@isy.liu.se).

F. Yin is with the Department of Science and Engineering, Chinese University of Hong Kong, Shenzhen, CN-518172, China. (E-mail: yinfeng@cuhk.edu.cn).

positioning using RSS, channel state information (CSI), TOA and time-difference-of-arrival (TDOA) has been provided in [4]. In [5], an overview of indoor positioning techniques from different dimensions have been provided, for instance, from camera to infrastructure based indoor positioning technology. Each has its own advantages based on the type of applications.

With the deployment of small cells, Wi-Fi access points and Bluetooth-low-energy (BLE) beacons, it is possible to utilize such infrastructure for positioning and thereby indoor location based services. A proximity measurement reveals whether or not a target of interest is in proximity (depending on the threshold) of a reference network node, for instance a radio base station, a BLE beacon or a Wi-Fi access point. Harness of proximity reports may result in new fashioned positioning systems with less signaling between the reporting device and core network, lower communication bandwidth, smaller database, as well as cheaper deployment and maintenance cost. Besides, there is a big trend nowadays in proximity related services and a big trend to see large scale, densely deployed, cooperative, hybrid BLE and WIFI networks under the umbrella of Internet of things (IoT), where we may easily measure a lot of RSS values from a big number of devices. Essentially, a proximity measurement is a quantized RSS with only two quantization levels (i.e., one bit) [6], [7]. Such kind of quantized data can be used in many applications. Since RSS measurements can be easily obtained from almost all kinds of devices, proximity measurements then are usually accessible.

# B. Related work and our contribution

The proximity measurement can be seen as inherited from such quantized data and one of the early examples was the positioning systems built with infrared (IR) sensors, e.g., [8]. The idea was also used in radio systems using mainly RSS as proximity indicator [9]. Furthermore, the threshold triggered report is also used as user equipment (UE) assisted measurement to support handover, for instance, in Wideband Code Division Multiple Access (WCDMA) and Long-Term Evolution (LTE) systems [10], [11]. In the literature, the positioning algorithms using proximity reports are called coarse grained algorithms or range-free algorithms. Since the publication of [9], a plethora of proximity report based positioning algorithms have been proposed, including the centroid algorithm [9], the approximate point in triangle (APIT) algorithm [12], the maximum-likelihood estimation algorithm [6], the ecolocation

1

algorithm [13], the iterative learning based algorithm [14], to mention a few. Most of them were designed for large-scale cooperative sensor network localization because of communication constraints, among others bandwidth and capacity limitations.

All previous approaches investigate static or snapshot solutions, such as fingerprinting methods for BLE based indoor positioning in [15] and [16]. Here instead, we propose particle filtering and smoothing. As compared to the static or snapshot solutions, particle filtering and smoothing are beneficial since they also consider the motion model of the device in the estimation process. Especially in cases where proximity reports are considered, the information contained in the measurement is limited and the motion model of the device will be helpful. The use of smoothing can be beneficial in many ways. While in filtering one estimates the state of the current time instance using measurements received up to that time instance, smoothing makes estimations at a certain time instance given all of the measurements up to a later time instance. The trajectory obtained in this way is usually smoother than the one using filtering. Besides, it is more likely to obtain better performance by taking advantage of the future measurements. The smoothed trajectories can be stored in the network and may be further used for some postprocessing and analysis whenever needed. Some examples of utilizing smoothed trajectories will be given later in Section 1.C. Compared with classical positioning methods such as Kalman filter (KF) and extended Kalman filter (EKF), particle filter and smoother generally have better performance when we have nonlinear/non-Gaussian models, which would be the case when proximity report is used. We also demonstrate this later in Section VI.

For static position estimation based on proximity report, two fundamental lower bounds have been proposed in [17], namely, the Barankin bound and Cramér-Rao bound (CRB). For nonlinear filtering/smoothing problems, parametric and posterior Cramér-Rao bounds (CRBs) are usually considered. Posterior CRB for non-linear filtering problems were discussed in [18] and [19]. The posterior CRB for quantized measurements has been derived in [20] and [21] but with different measurement models. Posterior CRBs can be interpreted as average bounds over all possible state (and measurement) sequences. However, with non-linear measurement model as investigated in this work, Monte Carlo averages are required to simulate (noisy) state and measurement sequence realizations (or trajectories) from the state-space model. In indoor pedestrian environments, it is practically impossible to generate trajectories which stay inside the building and which do not cross any walls. In order to overcome this practical limitation of posterior CRBs, parametric CRBs are considered in this paper, which put lower bounds on specific state sequence realizations. In [22], the general derivations of parametric CRB for filtering are given. To the best of our knowledge, the calculation of parametric CRBs have never been derived for proximity report based filtering and smoothing problems.

Our contributions of this paper are summarized as follows. In [1], a particle filtering algorithm has been developed for a proximity measurement model where the RSS threshold is selected in a more meaningful way as introduced in [23]. In this work, we further propose a particle smoothing algorithm for positioning based on the proximity reports. Evaluation results for static position estimation have also been included to motivate the advantages of using particle filtering and smoothing. Then, evaluations with the proposed algorithms are carried out with both simulated data as well as experimental data collected within an office area. In addition, the parametric CRBs for proximity report based filtering and smoothing are derived, which provide theoretical lower limits for the positioning performance under this framework.

# C. Main Motivation and Target Applications

The main target of this work is to develop a positioning scheme at a low cost, with reduced signaling between the user and network and possibly reduced power consumption at the user side. We aim at meter-level accuracy, which is sufficient for some room-level indoor positioning applications. To be more specific, the proximity reports have benefits over RSS reports when the user needs to send the report to a network server. This may happen for instance, when the network side needs to know the position of a certain device (rather than the device itself wants to obtain the position information). In such cases, the user needs to send periodic RSS reports to the network server. Then, by sending proximity reports instead, the user does not need to send actively, and may only report when a proximity event is triggered. Hence, the network can make use of the proximity reports sent from the user to estimate the position. This is a typical scenario where the proposed proximity mechanism can be applied. Such a way of sending the reports could potentially save energy on the user side, and also reduce signaling overhead between the user and network.

One potential usage of the proximity based positioning is to analyze and monitor different trajectories. It is also worth noting that the proposed proximity based solution fits well with different wireless infrastructures, especially BLE, which is usually densely deployed. Under dense deployment, proximity becomes more essential to reduce the communication resources. Considering the fluctuation of RSS in BLE systems, perform thresholding to obtain quantized proximity report is beneficial. To name a few examples, one may think of big shopping malls, which are equipped with BLE beacons, that could make use of this approach to estimate the position of customers with low cost. The shops then may use the position information to promote advertisements when there are customers close by. Besides, position information can be stored and further analyzed. Such post-processing will be useful to perform flow estimation during different time

Also, imagine that beacons can be placed at each showcase in a museum. By proximity detection and positioning, we can analyze the spatial-temporal trajectories of the visitors. The proximity based solution requires less communication resources. Hence, it is more suitable for museums such as British Museum, where there is a huge amount of visitors from globe. Trajectory analysis can be further used to understand how pedestrians visit the museum, which showcase attracts

most attention, and how to place the showcase such that it is more convenient for the visitors. Furthermore, those trajectories can be beneficial for the museum to monitor and detect abnormal behaviors, and even prevent terrorist attacks.

We may think about another practical example, where an indoor visual tracking system based on wireless sensor network is developed to perform tracking with high precision. The current visual detector relies heavily on the searching area of the moving target. If we can further refine it to a smaller area, the searching complexity of the visual tracking system will be greatly reduced. Here in this specific example, only coarse positioning is needed for refining the searching area. Besides, since refining the searching area is an auxiliary function, there may be limited resources allocated for this function. And the proposed proximity report based indoor positioning method fits well with this use case.

In the above mentioned applications, high accuracy positioning is not so essential. Especially thinking about the museum example, if other measurements, such as RSS, are used for positioning, the network server may quickly go down due to huge amount of data packages being transmitted. There are also other measurements for indoor positioning, which include but are not limited to TOA, CSI, and measurements from an inertial measurement unit (IMU). However, TOA requires precise synchronization, which is usually hard to achieve with current indoor infrastructure and also at a much higher cost. CSI can provide better positioning resolution. However, it requires perfect synchronization of timing, frequency and phase between the transmitter and receiver. Synchronization errors lead to contaminated CSI values. Besides, it is not always supported in all devices, while proximity can be easily obtained from all kinds of devices. There is also increasing popularity in an IMU based positioning scheme, since it can provide more accurate location estimates. However, this is realized in a more expensive way, especially when very high positioning accuracy is required (i.e., where lightweight IMU sensors with accurate measurements are needed).

## D. Paper organization and notations

The remainder of this paper is organized as follows: Section II gives detailed definition of the proximity report. Section III defines the state-space model for the problem at hand. Section IV introduces the novel proximity report based particle filtering and smoothing algorithms. Section V provides detailed derivation of theoretical parametric CRBs. Section VI validates the new proposed proximity report based particle filtering and smoothing algorithms and calculates the parametric CRBs in various experiments. Lastly, Section VII concludes the paper.

Throughout this paper, matrices are presented with uppercase letters and vectors with boldface lowercase letters. All the notations are defined and summarized in Table I.

#### II. PROXIMITY REPORTS

In this work, we consider a proximity report based method for indoor positioning, where model complexity appears as a

TABLE I NOTATIONS

Notation	Definition			
$[\cdot]^T$	Vector/matrix transpose			
$[\cdot]^{-1}$	Inverse of a non-singular square matrix			
$\operatorname{tr}(\cdot)$	Trace of a square matrix			
•	Euclidean norm of a vector			
•	Cardinality of a set			
$\mathbb{E}(\cdot)$	( )			
$X(\cdot)^T$	Short-hand notation for $XX^T$			
$\ln(\cdot)$	Natural logarithm			
$\log_{10}(\cdot)$	Logarithm to base 10			
8	Kronecker product			
$\nabla_{\boldsymbol{\theta}} = \partial/\partial \boldsymbol{\theta}$	Gradient operator			
$\Delta_{\boldsymbol{\theta}}^{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^{T}$	Laplace operator			
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean $\mu$ and variance $\sigma^2$			
$Cat(\cdot)$	Categorical distribution [24, page 35]			

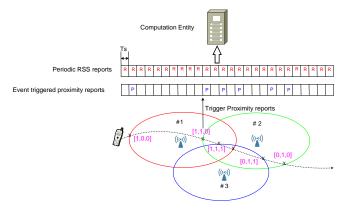


Fig. 1. Illustration of event triggered proximity reports.

hard-thresholding when converting an RSS measurement into a proximity measurement. More precisely,

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

It is noted that although hard-thresholding is considered in this work, it is also possible to generally model the thresholding as hysteresis. A proximity measurement obtained in the above way reveals whether or not a target of interest is in proximity of a *reference network node*. Given a vector of proximity measurements from different reference network nodes, the position of the device can be roughly estimated. However, the ambiguity given by such snapshot solution is still too large. In order to reduce the ambiguity, filtering and smoothing algorithms will be applied to proximity report based positioning.

Specifically, we consider centralized tracking of a mobile device. The mobile device measures RSS from reference network nodes that broadcast reference (i.e., pilot) signals periodically to the air. In the baseline centralized framework, the mobile device will upload the RSS measurements periodically to the *computation entity* in the core network either directly or via the reference nodes for position estimation. The computation entity is assumed to be equipped with sufficient computational power. Once the core network gets the position information of the mobile device, it can be used for assisting location based applications.

In order to reduce the signaling overhead between the mobile device and the computation entity, and the storage usage at the computation entity, an alternative is to only report a proximity measurement to the computation entity only when there is a change in the proximity measurement as shown in Fig. 1 (i.e., when a triggering condition such as equation (1) is met). The report generated in this way is defined as a proximity report. It needs to be emphasized that here we distinguish between a proximity measurement and a proximity report. The proximity measurement is generated each time when there is an RSS measurement. However, a proximity report (which contains the proximity measurement vector) is only triggered to be sent to the computation entity when there is a change in the proximity measurement (i.e., the mobile crosses a border when the RSS passes the threshold as shown by the cross marker in Fig. 1).

An illustrative example is also provided in Fig. 1. For instance, the proximity report containing the vector, e.g., [1,0,0], indicates that the UE is in the coverage area of the first reference node, while outside of the coverage area of the second and third reference nodes. The UE needs to upload the proximity report only if there is status change, for instance from [1,0,0] to [1,1,0], when the UE moved from left (i.e., in the coverage area of the first reference node) to the right into the coverage area of both first and second reference nodes.

In addition, instead of sending periodic RSS measurements, the device may send an RSS measurement vector of all observed reference network nodes (i.e., not only the node that triggered the proximity measurement) only on triggering of such event. However, in this paper, we only study and compare the positioning performance of the periodic RSS and event-triggered proximity reports. Positioning performance of such event-triggered RSS measurements have been studied in [25].

#### III. SYSTEM MODEL

We consider an indoor sensor network which comprises  $N_B$  reference network nodes with a priori known positions,  $\mathbf{p}_j$ ,  $j=1,2,\ldots,N_B$ , a central computation entity and a mobile device, whose state  $\boldsymbol{x}_k$  is to be estimated at each time step k from a set of available measurements. To that end, the mobile device collects RSS measurements, converts them to proximity measurements, and further reports them to the computation entity, when it is triggered for position estimation.

The mobile device estimation problem is cast into a statespace model formulation, which, in its most general form, is given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k), \tag{2a}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{e}_k), \tag{2b}$$

where  $y_k$  denotes the measurement vector,  $w_k$  is the process noise,  $e_k$  is the measurement noise, and  $f_k(\cdot)$ ,  $h_k(\cdot)$  are arbitrary linear/nonlinear mappings of appropriate size. Here, the process model (2a) accounts for modeling the motion of pedestrians in indoor environments.

In the literature, several motion models have been suggested for indoor pedestrian movements, for instance, random walk [26], constant velocity [27] or nearly constant velocity [28].

These models give different transitions from the previous state to the current state and different process noise covariances. In this work, considering the relative static indoor environment and low walking speed, we use the white noise nearly constant velocity model as given in [29, Chapter 6]. The corresponding process model is given by

$$\boldsymbol{x}_{k+1} = F\boldsymbol{x}_k + \boldsymbol{w}_k, \tag{3}$$

with state vector  $\boldsymbol{x}_k = [p_{x,k}, v_{x,k}, p_{y,k}, v_{y,k}]^T$ , and where  $\bar{\boldsymbol{p}}_k = [p_{x,k}, p_{y,k}]$  and  $[v_{x,k}, v_{y,k}]$  denote the 2-D position and velocity, respectively. The transition matrix F is given by

$$F = I_2 \otimes \left[ \begin{array}{cc} 1 & T_s \\ 0 & 1 \end{array} \right], \tag{4}$$

where  $I_2$  is the identity matrix of size 2, and  $T_s$  is the sampling time. The vector  $w_k$  is assumed to be a zero-mean Gaussian distributed with covariance matrix

$$\Sigma_w = I_2 \otimes \begin{bmatrix} \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s}{2} & T_s \end{bmatrix} \sigma_w^2, \tag{5}$$

where  $\sigma_w^2$  denotes the noise power spectral density.

The measurement model aims at relating the measurements to the unknown state, which is to be estimated. In the sequel, we review the conventional RSS model and introduce the proximity measurement model. Here it is usually assumed that the mean values of RSS stay almost unaltered with time, which is generally true when the indoor propagation environment is not changing significantly, for instance, there is no significant change in the layout, such as removing or adding walls, changing furniture placement, and so on. To support this assumption, some experimental results are included in Section 1 of a supplementary report [30]. However, it should be noted that when the radio propagation condition (environment) is changing significantly with time, the model can be easily updated accordingly. The magnitude of RSS measurements usually decays logarithmically with distance (between a reference node and the mobile device) [31]. This is conventionally modeled using a log-distance model, which is given by

$$r_{k,j} = \underbrace{A_j + 10B_j \log_{10} \left(\frac{d_{k,j}}{d_0}\right)}_{\mu_{k,j}} + e_{k,j},$$
 (6)

where  $r_{k,j}$  is the RSS measurement (in dBm) from the jth reference network node at time step k and  $A_j$  denotes the received power (in dBm) measured at a reference distance  $d_0$ .  $B_j$  (in dB) is the (negative) path loss exponent, and  $d_{k,j} \triangleq ||p_j - \bar{p}_k||$ . The measurement noise  $e_{k,j}$  (in dB) accounts for unmodeled errors and is usually assumed to be zero-mean Gaussian distributed with variance  $\sigma_i^2$ .

The propagation model parameters  $A_j$ ,  $B_j$ , and  $\sigma_j^2$  are generally unknown, but can be calibrated in an offline phase given a batch of RSS measurements collected from all reference nodes [32]. In the present work we assume that these parameters are known. In indoor environments, the path loss exponent  $B_j$  is typically ranging from -3.5 dB to -1.6 dB [33, Chapter 2]. In the following, we assume that the

RSS measurements collected for different reference nodes are mutually independent. This yields

$$\mathbf{y}_{\mathrm{R},k} = \mathbf{h}(\mathbf{x}_k) + \mathbf{e}_k,\tag{7}$$

where  $y_{R,k} = [r_{k,1}, \dots, r_{k,N_B}]^T$  and  $h(x_k) = [\mu_{k,1}, \dots, \mu_{k,N_B}]^T$ . The model for the proximity measurements is obtained by hard-thresholding each RSS measurement, according to (1), yielding

$$\mathbf{y}_{\mathrm{P},k} = \mathbf{g}(\mathbf{y}_{\mathrm{R},k}) \tag{8}$$

where  $\mathbf{y}_{\mathrm{P},k} = [y_{k,1},\ldots,y_{k,N_B}]^T$  is the vector of proximity measurements,  $\mathbf{g}(\mathbf{y}_{\mathrm{R},k}) = [g(\mu_{k,1} + e_{k,1}),\ldots,g(\mu_{k,N_B} + e_{k,N_B})]^T$  and g(x) is a non-linear function which performs hard-thresholding of an input, x, as follows:

$$g(x) = \begin{cases} 0, & x \le P_{\text{th}}, \\ 1, & x > P_{\text{th}}. \end{cases}$$
 (9)

Here, it is worth recalling that proximity measurements are *not* reported to the computational entity at every time step, but only triggered when there is a change in the proximity measurement. In terms of notation, we assume that the sequence of periodically transmitted RSS measurements and the sequence of proximity reports from time step 1 up to time  $\ell$  is defined as  $y_{1:\ell} = [y_1^T, \ldots, y_\ell^T]^T$ , i.e. both sequences underlie the same (constant) sampling period  $T_s$ . In case proximity reports have not been received at a specific time step m, the corresponding vector  $y_m$  is empty. However, this empty vector still contains information. It can be inferred that the proximity measurement does not change since the previous report.

# IV. BAYESIAN FILTERING AND SMOOTHING

In Bayesian filtering, we are interested in estimating the state  $x_k$  at each time step, given the history of measurements  $y_{1:k}$  up to time step k. In probabilistic terms, this amounts to extract information from the posterior density  $p(x_k|y_{1:k})$ , such as the (posterior) mean. Conceptually, the posterior density  $p(x_k|y_{1:k})$  can be updated recursively, yielding the so-called Bayesian filtering equations, see e.g. in [34, Chapter 1]. Closed-form solutions to this recursion are available only in a few special cases, e.g. in case the state-space model is linear, additive Gaussian, the recursion is given by the celebrated Kalman filter [35].

In Bayesian smoothing, we are interested in estimating the state  $x_k$  of each time step, given that we have received a set of measurements  $y_{1:T}$ , where T > k. The corresponding posterior density  $p(x_k|y_{1:T})$  again can be conceptually computed recursively, using e.g. the forward-backward (or simply Bayesian) smoothing equations, with closed-form solutions, also known as the Kalman smoother, in the case of linear, additive Gaussian state-space models [36]. The state-space model proposed in Section III, however, does not admit the application of the Kalman filter or smoother, since the measurements are highly nonlinear, and we therefore have to resort to approximations.

In case of nonlinear state-space models, a plethora of algorithms have been proposed in the literature to approximate

the Bayesian filtering and smoothing equations [34]. In the following, we apply sequential Monte Carlo methods, concretely particle filtering and smoothing, to numerically approximate the posterior densities of interest.

#### A. Particle Filtering

Particle filtering aims at numerically approximating the posterior density  $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$  by a finite set of N weighted samples (also called particles) that evolve dynamically on a probabilistic grid of the state-space. It is based on sequential importance sampling, and became practically useful after adding a resampling step [37]. Since its development in the early 1990s, various particle filter variants have appeared in the literature [38]. In this paper, we focus on the bootstrap particle filter, with importance density chosen to be the transitional prior  $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k; F\boldsymbol{x}_{k-1}, \Sigma_w)$ , because it is simple and provided satisfactory performance for our application. The bootstrap particle filter, tailored to our estimation problem is summarized in Algorithm 1.

# Algorithm 1 Bootstrap Particle Filter

- 1) Initialization: Draw N samples  $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$ , and set  $w_0^{(i)} = 1/N$ , for all i=1,...,N.
- 2) For k = 1, ..., T, do:
  - a) Draw N samples  $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}),$  for i=1,...,N.
  - b) Update weights according to

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

for i = 1, ..., N, where

- Proximity:  $p(y_k|x_k^{(i)}) = p(y_{P,k}|x_k^{(i)})$  as defined in (10).
- RSS:  $p(y_k|x_k^{(i)}) = p(y_{R,k}|x_k^{(i)})$  as defined in (13).
- c) For i = 1, ..., N, normalize the weights to sum to unity,

$$w_k^{(i)} = \frac{\widetilde{w}_k^{(i)}}{\sum_{i=1}^N \widetilde{w}_k^{(i)}}.$$

 d) An approximation to the posterior filtering expectation is given by

$$\hat{oldsymbol{x}}_{ ext{PF},k}(oldsymbol{y}_{1:k}) \stackrel{\Delta}{=} \sum_{i=1}^N w_k^{(i)} oldsymbol{x}_k^{(i)} pprox \mathbb{E}\{oldsymbol{x}_k | oldsymbol{y}_{1:k}\}.$$

e) Resampling: If  $N_{\rm eff} = 1/\sum_{i=1}^N (w_k^{(i)})^2 < N_{\rm th}$ , then perform multinomial resampling [39].

Note that at each time step, the weights will be updated, see step 2-b in Algorithm 1. In case there is a proximity report available, the weights will be computed based on the available proximity report. When there is no proximity report received at the computation entity (i.e., the corresponding vector is empty), it indicates the proximity measurement stays unchanged since the last time when a proximity report was received. In such cases, the weights will still be updated using

the most recently received proximity report. The likelihood function is evaluated as follows:

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

$$= \prod_{j=1}^{N_B} \sum_{l \in \{0,1\}} \delta(y_{k,j} - l) p(y_{k,j} = l|\mathbf{x}_k), \quad (10)$$

where  $\delta(\cdot)$  denotes the Dirac delta function. It is easy to prove further that

$$p(y_{k,j} = 0 | \boldsymbol{x}_k) = \Phi\left(\frac{P_{\text{th}} - \mu_{k,j}}{\sigma_j}\right),\tag{11a}$$

$$p(y_{k,j} = 1|\mathbf{x}_k) = 1 - p(y_{k,j} = 0|\mathbf{x}_k),$$
 (11b)

where

$$\Phi\left(\frac{t-\mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{t} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] dx. \quad (12)$$

Correspondingly, the likelihood function for RSS measurements is calculated as

$$p(\mathbf{y}_{R,k}|\mathbf{x}_k) = \prod_{j=1}^{N_B} p(r_{k,j}|\mathbf{x}_k)$$

$$= \prod_{j=1}^{N_B} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(r_{k,j} - \mu_{k,j})^2}{2\sigma_j^2}\right]. \quad (13)$$

#### B. Particle Smoothing

In particle smoothing, we aim at approximating the posterior  $p(x_k|y_{1:T})$  by a set of weighted particles. Compared to the particle filter, which is an online method as it processes measurements up to the current time  $y_{1:k}$ , the particle smoother performs state inference (offline) on the entire set of measurements  $y_{1:T}$  collected during the experiment. In the literature, various particle smoothing algorithms have been proposed [36], [40], [41]. In this work, we have implemented the Forward Filtering Backward Simulation (FFBSi) smoother [36] due to its simplicity and satisfactory performance. The FFBSi smoother is a forward-backward filter, where in the forward pass an ordinary particle filter is used, and in the backward pass individual state trajectories are sampled to obtain a satisfactory representation of the smoothing posterior density. The FFBSi smoother is summarized in Algorithm 2.

## V. PARAMETRIC CRAMÉR-RAO LOWER BOUNDS

Assessing the fundamental performance limitations in Bayesian filtering and smoothing is usually carried out using the posterior (or Bayesian) Cramér-Rao bound (CRB). In Bayesian filtering, the posterior CRB  $B_{\text{filt}}$  puts a lower bound on mean square error (MSE) matrix of any estimator  $\hat{x}_k(y_{1:k})$ , i.e.

$$\mathcal{M}(\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k})) \stackrel{\Delta}{=} \mathbb{E}\left\{ (\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k}) - \boldsymbol{x}_k)(\cdot)^T \right\} \succeq B_{\text{filt}}, \quad (14)$$

where the inequality means that the difference  $\mathcal{M}(\hat{x}_k(y_{1:k}))$  –  $B_{\text{filt}} \succeq 0$  is positive semi-definite. Similarly, for the posterior CRB of Bayesian smoothing  $B_{\text{smooth}}$ , we have for any estimator  $\hat{x}_k(y_{1:T})$  the relation  $\mathcal{M}(\hat{x}_k(y_{1:T})) \succeq B_{\text{smooth}}$ . Posterior

#### Algorithm 2 FFBSi Smoother

- Run bootstrap particle filter according to Algorithm 1. Store all particles and corresponding weights as  $\left\{ \{w_k^{(i)}\}_{i=1}^N, \{x_k^{(i)}\}_{i=1}^N \right\} \text{ for } k = 1, ..., T.$
- 2) At time T, sample independently  $\{Id(m)\}_{m=1}^{M_{\rm BS}} \sim \operatorname{Cat}\left(\{w_T^{(i)}\}_{i=1}^N\right)$  and set  $\widetilde{\boldsymbol{x}}_T^{(m)} = \boldsymbol{x}_T^{(Id(m))}$ , for m=1
- 3) For each  $\ell = T 1, ..., 1$ , do the following:
  - a) For  $m = 1, ..., M_{BS}$ , do the following:
    - i) Calculate new weights according to

$$\widetilde{w}_{\ell\mid T}^{(i,m)} \propto w_{\ell}^{(i)} p(\widetilde{\boldsymbol{x}}_{\ell+1}^{(m)}|\boldsymbol{x}_{\ell}^{(i)}), i = 1, ..., N.$$

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

$$w_{\ell|T}^{(i,m)} = \frac{\widetilde{w}_{\ell|T}^{(i,m)}}{\sum_{i=1}^{N} \widetilde{w}_{\ell|T}^{(i,m)}}.$$

$$\begin{array}{lll} \text{iii)} & \operatorname{Draw} \, Id(m) \sim \operatorname{Cat} \left( \{ w_{\ell \mid T}^{(i,m)} \}_{i=1}^{N} \right). \\ \text{iv)} & \operatorname{Set} \quad \widetilde{\boldsymbol{x}}_{\ell}^{(m)} & = & \boldsymbol{x}_{\ell}^{(Id(m))}, \quad \text{and} \quad \widetilde{\boldsymbol{x}}_{\ell:T}^{(m)} & = \\ \left\{ \widetilde{\boldsymbol{x}}_{\ell}^{(m)}, \widetilde{\boldsymbol{x}}_{\ell+1:T}^{(m)} \right\}. \end{array}$$

b) An approximation to the posterior smoothing expectation is given by

$$\hat{m{x}}_{ ext{PS},\ell}(m{y}_{1:T}) \stackrel{\Delta}{=} rac{1}{M_{ ext{BS}}} \sum_{m=1}^{M_{ ext{BS}}} \widetilde{m{x}}_{\ell}^{(m)} pprox \mathbb{E}\{m{x}_{\ell} | m{y}_{1:T}\}.$$

CRBs can be interpreted as average bounds over all possible state (and measurement) sequences. In case of linear, additive Gaussian state-space models, the posterior CRBs reduce to the (closed-form) covariance recursions of the Kalman filter and smoother, and hence are simple to compute. The estimation problem investigated in this paper, however, is nonlinear and requires to compute expectations w.r.t. state and measurement sequences that are not available in closed-form, see [42] for details. Hence, for computing the bounds it is required to form Monte Carlo averages by simulating (noisy) state and measurement sequence realizations (or trajectories) from the state-space model. While this seems mathematically appealing, it results in that bounds are computed for state trajectories that not necessarily reflect indoor pedestrian motion. In particular, with the simple model defined in (3), it is practically impossible to generate trajectories which stay inside the building and which do not cross any walls.

In order to overcome this practical limitation of posterior CRBs, parametric CRBs for Bayesian filtering and smoothing have been recently introduced. Parametric CRBs aim at bounding from below the MSE matrix conditioned on a specific state sequence realization  $x_{0:k}^*$  or  $x_{0:T}^*$ . Due to the conditioning, there is no need to generate state trajectories from the process model, and one can use instead any state trajectory of interest.

## A. Parametric CRB for Filtering

The parametric CRB for filtering  $P_{k|k}(\boldsymbol{x}_{0:k}^*)$  puts a lower bound on the conditional MSE matrix for any unbiased estimator  $\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k})$  as follows

$$\mathcal{M}(\hat{\boldsymbol{x}}_{k}(\boldsymbol{y}_{1:k}) \mid \boldsymbol{x}_{0:k}^{*}) = \mathbb{E}\{(\hat{\boldsymbol{x}}_{k}(\boldsymbol{y}_{1:k}) - \boldsymbol{x}_{k}^{*})(\cdot)^{T} \mid \boldsymbol{x}_{0:k}^{*}\}$$

$$\succeq P_{k|k}(\boldsymbol{x}_{0:k}^{*}), \tag{15}$$

where  $P_{k|k}(\boldsymbol{x}_{0:k}^*)$  is the inverse of an auxiliary Fisher information matrix (FIM) [22]. The auxiliary FIM can be computed recursively, and, similarly to the posterior CRB, for the case of linear, additive Gaussian state-space models, it is related to the covariance recursions of the Kalman filter, but where the process noise covariance matrix is set to zero [22]. The computation of the parametric filtering CRB adapted to our proximity report (and RSS) based estimation problem is given in Algorithm 3.

It is worth noting that the contribution of the proximity

# Algorithm 3 Parametric Filtering CRB

- 1) Initialize the FIM with  $J_{0|0}(\boldsymbol{x}_0^*) = \Sigma_{\boldsymbol{x}_0}^{-1}$ , in case the prior is Gaussian  $p(\boldsymbol{x}_0^*) = \mathcal{N}(\boldsymbol{x}_0^*; \mu_{\boldsymbol{x}_0}, \Sigma_{\boldsymbol{x}_0})$ . Compute the parametric filtering CRB as  $P_{0|0}(\boldsymbol{x}_0^*) = J_{0|0}^{-1}(\boldsymbol{x}_0^*)$ .
- 2) For k = 1, ..., T, do:
  - Compute the filtering FIM from

$$J_{k|k} = \left[ F J_{k-1|k-1}^{-1} F^T \right]^{-1} + H_k^T(\boldsymbol{x}_k^*) R_k^{-1} H_k(\boldsymbol{x}_k^*)$$

where  $J_{k|k}$  is a short-hand notation for  $J_{k|k}(\boldsymbol{x}_{0:k}^*)$ , and where  $H_k^T(\boldsymbol{x}_k^*)R_k^{-1}H_k(\boldsymbol{x}_k^*)$  is given by

- Proximity: (29),
- RSS: (27).
- Compute the parametric filtering CRB as  $P_{k|k}(\boldsymbol{x}_{0:k}^*) = J_{k|k}^{-1}(\boldsymbol{x}_{0:k}^*).$

measurement, i.e.  $H_k^T(\boldsymbol{x}_k^*)R_k^{-1}H_k(\boldsymbol{x}_k^*)$ , is updated at each time step even when there is no proximity report received at the computation entity (i.e., the corresponding measurement vector is empty). This is done because no triggering of proximity report still contains information that the proximity measurement stays unchanged since the last time a proximity report was received. Detailed derivations of the parametric filtering CRB can be found in [22]. The adaptions performed to the bound to hold for proximity and RSS measurements are detailed in the Appendix.

# B. Parametric CRB for Smoothing

The parametric CRB for smoothing  $P_{k|T}(\boldsymbol{x}_{0:T}^*)$  provides a lower bound on the MSE matrix of any unbiased estimator  $\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:T})$  and can be stated as follows

$$\mathcal{M}(\hat{x}_k(y_{1:T}) \mid x_{0:T}^*) \succeq P_{k|T}(x_{0:T}^*).$$
 (16)

The inverse of  $P_{k|T}(\boldsymbol{x}_{0:T}^*)$  is related to the FIM of the smoothing problem and can be computed recursively using a forward-backward procedure as is shown in [43]. Again, in the case of a linear, additive Gaussian state-space model, the parametric CRB for smoothing is related to the covariance recursions of the Kalman smoother, but with zero process noise covariance matrix. The computation of the parametric CRB for smoothing is summarized in Algorithm 4.

# Algorithm 4 Parametric Smoothing CRB

- 1) For k = 1, ..., T, calculate the filtering FIM  $J_{k|k}$  as given in Algorithm 3.
- 2) For  $\ell = T 1, ..., 1$ , do:
  - Compute the smoothing FIM from

$$J_{\ell|T} = J_{\ell|\ell} + F^T \left[ J_{\ell+1|T} - \left[ F J_{\ell|\ell}^{-1} F^T \right]^{-1} \right] F.$$

where  $J_{\ell|T}$  is a short-hand notation for  $J_{\ell|T}(x_{0:T}^*)$ .

• Compute the parametric smoothing CRB as  $P_{\ell|T}(x_{0:T}^*) = J_{\ell|T}^{-1}(x_{0:T}^*).$ 

#### VI. RESULTS

# A. Experimental Setup

The proposed proximity based indoor positioning algorithm is evaluated under different layouts, for instance, an office layout and an open area. Due to space limitation, results for the office layout is provided here. For evaluations conducted in an open area, please consult to Section 3.2 in the supplementary report [30]. We consider a typical office environment at Ericsson, Linköping, Sweden. In total  $N_{\rm B}=10$  BLE beacons are placed rather uniformly in the area such that a generally good geometry is obtained. For instance, as mentioned in [44], if the geometric dilution of precision (GDOP) value is within a certain range, the positioning accuracy will not differ too much from the case with optimal placement. The GDOP for this deployment is computed and plotted in Section 2 of the supplementary report [30]. The GDOP value for this deployment is below 1 for most positions, and below 3 for all positions, which indicates an overall good geometry [45].

The floor plan as well as the known beacon positions are shown in two-dimensional (2-D) space in Section 3.1 of the supplementary report [30], where, a local coordinate system is used. It should be emphasized that in this paper, the study of optimal beacon placement and the effect of beacon density is not investigated. However, it would be an interesting direction of our future research work. The BLE beacons serve as transmitters and broadcast information regularly. A moderate scale measurement campaign was conducted during normal work hours. Throughout the measurement campaign, the mobile device (equipped with BLE chipset) monitor the BLE advertisement channel and measures RSS. A total of M = 12144 RSS measurements were collected offline along 52 predefined tracks, from which a set of propagation parameters  $\{A_j, B_j, \sigma_j^2\}_{j=1}^{N_{\rm B}}$  were estimated for each BLE beacon using standard (batch) linear least squares (LLS) regression method [46]. If there is any significant change in the indoor environment, a new set of training data needs to be collected and all propagation parameters need to be updated. This is a classical way of training propagation parameters which can be found for instance in [33, Section 2.8] and [47].

After this offline training phase, the propagation parameters are used together with the RSS propagation model to calculate the optimal RSS threshold  $P_{\rm th}$  according to [23]. 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 [23].

After the propagation parameters as well as the threshold are obtained, the evaluation of the proposed particle filtering and smoothing algorithms are performed. In this work, two scenarios are considered. In the first scenario, we simulate RSS and proximity report measurements along selected state trajectories using the measurement models introduced in Section III, with propagation parameters and optimal RSS threshold estimates obtained from the offline training phase. The achieved positioning performance of the particle filter and smoother is then compared to the corresponding theoretical performance limits via the parametric CRB. The second scenario investigates the performance of the proposed algorithms using real RSS measurements provided by the mobile device, as well as proximity reports which are obtained in a post-processing step by comparing RSS with the optimal threshold.

## B. Performance Metrics

In order to compare the positioning accuracy of the proposed filtering and smoothing algorithms in indoor environments, the following performance metrics are introduced. The conditional position estimation error at time k is calculated as follows:

$$E_{k|\tau,m} = ||\hat{\boldsymbol{p}}_{k|\tau}^{(m)} - \bar{\boldsymbol{p}}_{k}^{*}||, \tag{17}$$

where  $\hat{p}_{k|\tau}^{(m)} = [\hat{p}_{\mathbf{x},k|\tau}^{(m)},\hat{p}_{\mathbf{y},k|\tau}^{(m)}]^T$  are the position estimates of the estimator  $\hat{x}_k(\mathbf{y}_{1:\tau}^{(m)})$  obtained from the m-th Monte Carlo run (filtering:  $\tau = k$ , smoothing:  $\tau = T$ ), and  $\bar{p}_k^* = [p_{\mathbf{x},k}^*, p_{\mathbf{y},k}^*]^T$  denote the ground truth. Here, the term conditional is referred to the fact that each measurement sequence  $\mathbf{y}_{1:\tau}^{(m)}$ ,  $m = 1, \ldots, M$  has been simulated using the same state vector realization  $\mathbf{x}_{0:\tau}^*$ . Given M Monte Carlo runs, the position root conditional MSE (RCMSE) can be approximated by

$$RCMSE_{k|\tau} \approx \sqrt{\frac{1}{M} \sum_{m=1}^{M} (E_{k|\tau,m})^2}.$$
 (18)

The position RCMSE will be compared to the parametric CRB. In case the estimator  $\hat{x}_k(y_{1:\tau})$  is conditionally unbiased, the following relation holds

$$\text{RCMSE}_{k|\tau} \ge \sqrt{P_{k|\tau}^{1,1} + P_{k|\tau}^{3,3}} \triangleq \gamma_{k|\tau},$$
 (19)

where  $P_{k|\tau}^{i,j}$  denotes the element at row i column j of the parametric CRB matrix  $P_{k|\tau}(\boldsymbol{x}_{0:k}^*)$ . The parametric CRB presented in this work requires that the estimator is conditionally unbiased, a condition that is rarely met in practice by any Bayesian estimator. It is therefore instructive to assess the estimator's conditional bias to correctly interpret the achieved results. The conditional position bias vector can be approximated as follows:

$$b_{k|\tau} \approx \frac{1}{M} \sum_{m=1}^{M} \left[ \hat{p}_{k|\tau}^{(m)} - \bar{p}_{k}^{*} \right],$$
 (20)

where  $b_{k|\tau} = [b_{x,k|\tau}, b_{y,k|\tau}]^T$ . Lastly, in order to get an overall positioning accuracy in the whole office area, we define the RCMSE across various state trajectories. It is given by

$$\overline{\text{RCMSE}}_{k|\tau} \approx \frac{1}{N_t} \sum_{t=1}^{N_t} \text{RCMSE}_{k|\tau}^{(t)}, \tag{21}$$

where  $N_t$  is the number of evaluated state trajectories and  $\text{RCMSE}_{k|\tau}^{(t)}$  is the RCMSE obtained for the t-th state trajectory. Correspondingly, the following relation holds for conditionally unbiased estimators

$$\overline{\text{RCMSE}}_{k|\tau} \ge \frac{1}{N_t} \sum_{t=1}^{N_t} \gamma_{k|\tau}^{(t)}, \tag{22}$$

where  $\gamma_{k|\tau}^{(t)}$  is the bound for t-th trajectory and  $\frac{1}{N_{\rm t}}\sum_{t=1}^{N_{\rm t}}\gamma_{k|\tau}^{(t)}$  computes the average bound for different state trajectories of the same length. Finally, we define the average RCMSE as

$$\overline{\text{RCMSE}} \triangleq \frac{1}{T} \sum_{k=1}^{T} \overline{\text{RCMSE}}_{k|\tau}.$$
 (23)

TABLE II EVALUATION PARAMETERS.

Parameter	Value	Description	
$\sigma_w$	1	Noise power spectral density	
$T_{ m s}$	0.1 sec	Sampling interval	
$P_{th}$	−82 dBm	Predefined RSS threshold	
N	2000	Number of particles	
$N_{ m B}$	10	Number of deployed reference nodes	
$\mu_{x_0}$	$[15.5, 1.5, 19.5, 1.5]^T$	Mean of $\boldsymbol{x}_0$	
$\Sigma_{x_0}$	diag([1, 2, 1, 2])	Covariance of $x_0$	
$N_{ m th}$	2/3 N	Threshold for resampling	
$M_{ m BS}$	10	Number of FFBSi backward trajectories	

#### C. Simulated Data

In this section, the proposed algorithms are evaluated with simulated data. With propagation parameters estimated from the real data as described in Section VI-A and  $P_{th}$  preselected, RSS and proximity reports are simulated along selected state trajectories using the measurement models given in Section III. With simulated data, the measurement vector contains values from all beacons, which may not be the case in practice. Then, combining the motion model and simulated measurements, particle filtering and smoothing have been applied to obtain position estimates. The parameters specified for particle filtering and smoothing are listed in Table II. The sampling interval  $T_s$  is set to 0.1 second, which corresponds to the advertisement interval of BLE beacon used in this study. It should be noted that since the measurements are simulated, it is the ideal case that we can assume RSS is measured regularly at fixed sampling interval  $T_s$ . In the motion model,  $\sigma_w$  is a tuning parameter. Various values have been tested and  $\sigma_w = 1$  yields good results. The number of particles is set to N=2000, since no significant difference is observed with N > 1000. For FFBSi smoothing, the number of backward trajectories simulated is selected as  $M_{\rm BS}=10$ , considering both the computation complexity and no significant improvement for  $M_{\rm BS} > 10.$ 

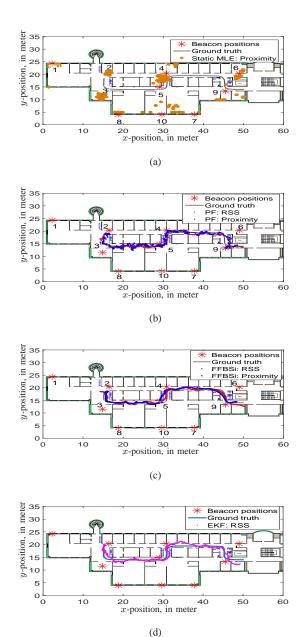


Fig. 2. Illustration of position estimates and ground truth with simulated data: 2(a) static MLE, 2(b) PF, 2(c) FFBSi, 2(d) EKF.

To motivate the advantages of applying particle filtering and smoothing, an illustration of the static position estimates using maximum likelihood estimator (MLE) in [17] is shown in Fig. 2(a) with one specific state trajectory. 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 restroom) and consists of a number of 90 degrees turns.

In order to find the optimal position estimate which maximizes the likelihood function, a 2-D grid search method with resolution of 0.1 meter has been applied. From the results we can see that the estimated position in this static case is not satisfactory, which motivates the need for particle filtering and smoothing approaches. The positioning ambiguity is very large in static case, since the location information provided by

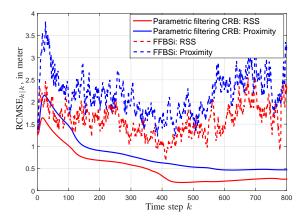


Fig. 3.  $RCMSE_{k|k}$  and parametric filtering CRB.

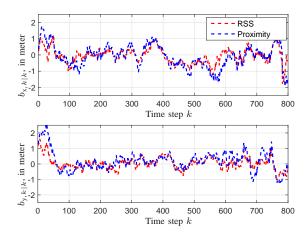


Fig. 4. Bias in position estimates for filtering.

proximity reports is very limited. This also infers that when the information from the measurement is little, such as in this case with binary proximity reports, the motion model is of great importance to get better positioning accuracy, which will be demonstrated later in this section.

Positioning results with particle filtering and smoothing are plotted on Fig. 2(b) and 2(c), respectively. Compared with the static position estimates, significant improvements are observed. In addition, the algorithm based on proximity report exhibits similar performance as compared to algorithm based on RSS measurements. The first couple of 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 proximity of all the beacons.

Extended Kalman filter (EKF) is always considered as an alternative method to solve nonlinear filtering problems. For RSS measurements with EKF implementation, relevant research work can be found, for instance, in [48], [49] and [50]. Here in this paper, we also compare the proposed particle filtering algorithm with EKF. The tuning parameter for the process noise is set to  $\sigma_w=0.3$ , which yields relatively

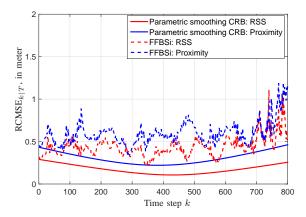


Fig. 5.  $RCMSE_{k|T}$  and parametric smoothing CRB.

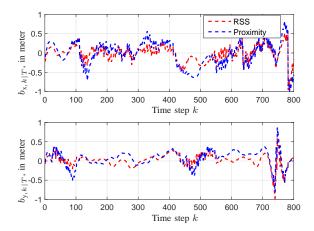


Fig. 6. Bias in position estimates for smoothing.

better performance. From results in Fig. 2(d) and Fig. 8, it is clear that PF outperforms EKF for RSS measurements. To the best of the author's knowledge, a proximity-based EKF has not been proposed in the literature and is subject to future research.

To evaluate the theoretical limits of the positioning performance, the RCMSE is compared with the parametric CRBs calculated in Section V. The number of Monte Carlo runs is set to M=10000. The parametric CRB for particle filtering and smoothing of the state trajectory in Fig. 2 is plotted in Fig. 3 and Fig. 5. It is observed that the position error appears above the parametric bound, showing the positioning performance varies spatially over the office floor and in relation to the beacon deployment. Bias in the positioning estimates are observed and plotted in Fig. 4 and Fig. 6. It is noticed that the RCMSE goes further away from the parametric CRB at the end of the trajectory, this may be due to the larger bias accumulated after k=400.

In order to get an overall positioning accuracy in the office area, we compute the average parametric CRBs in (21) for  $N_{\rm t}=5$  state trajectories across the deployment area of the same length. The average CRBs are compared with  $\overline{\rm RCMSE}_{k|\tau}$  in (22) in Fig. 7. It can be seen that the average position performance of the office is similar to the positioning

performance of the selected trajectory in Fig. 2. This is due to the fact that the office is mainly characterized by two corridors and a shape of a narrow rectangle. Therefore, the selected trajectory captures most of the characteristics of the office.

The overall CDF of position errors of those different state

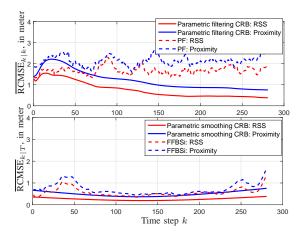


Fig. 7. Average positioning accuracy.

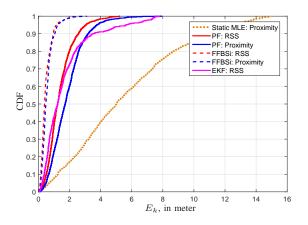


Fig. 8. CDF of position errors with simulated data.

TABLE III
POSITION ERRORS (IN UNIT OF METER) AT DIFFERENT PERCENTILES.

Percentile		50-th	67-th	95-th
Simulated	Static MLE(Proximity)	5	7	12.5
	PF(RSS/Proximity)	1.2/1.7	1.6/2.2	3/3.7
	FFBSi(RSS/Proximity)	0.4/0.5	0.5/0.6	1.1/1.3
Experimental	Static MLE(Proximity)	6.2	7.8	12.2
	PF(RSS/Proximity)	2.5/3.5	3.2/4.6	5.5/8.1
	FFBSi(RSS/Proximity)	2.3/3	3.1/4.2	4.8/6.6

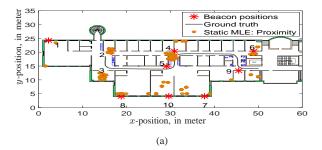
trajectories have been plotted in Fig. 8. The 50-th, 67-th and 95-th percentile of the position error are listed in Table III. It is noted that for PF based on proximity report, there is approximately 0.6 meter degradation as compared with PF based on RSS. However, by applying FFBSi smoothing, the performance difference between proximity report and RSS based methods are negligible.

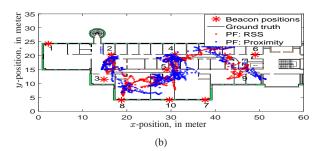
## D. Experimental Data

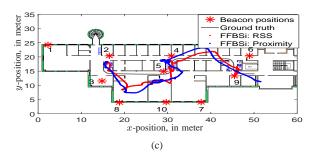
In the above simulations, it is assumed that RSS measurements from all beacons in the deployment area are available to the mobile device. However, in practical environments, it is not realistic since the device may only measure RSS from 3 or 4 beacons at one position. In what follows, in order to show performance of the proposed algorithms in practical circumstances, evaluations based on real RSS measurements carried out in the office area are provided. All parameters for particle filtering and smoothing are listed in Table II. In real circumstances, the BLE advertisement interval is 100 ms plus or minus 0-10 ms. Hence, the RSS may not be measured with the exactly fixed interval. In the first case studied here, a regular sampling interval of 100 ms is assumed to ease the implementation. With sampling interval set to 100 ms, the instanuous RSS measurement is used without any averaging, since only one sample can be obtained during this interval. In the second case, we will also evaluate the positioning performance using a different sampling interval such that RSS averaging is possible. It needs to be mentioned that the number of particles for experimental data is also set to N = 2000, since no significant improvement can be observed for a larger number. The ground truth positions are obtained from an appbased positioning algorithm developed by Senion.

The static position estimates obtained from MLE are depicted in Fig. 9(a). The position estimates in static case are not satisfactory and filtering and smoothing are needed. Fig. 9(b) and 9(c) show positioning performance with particle filtering and smoothing, respectively. Similar performance is still achieved for proximity report and RSS based positioning. The particle filtering and smoothing algorithms work well in distinguishing if the device is in the upper or lower corridor area. By applying FFBSi smoothing, the positioning accuracy is improved for both RSS and proximity reports. The performance with EKF for RSS is plotted in Fig. 9(d). The tuning parameter  $\sigma_w$  in process model is also set to 0.3 for experimental data. Obviously, EKF performs worse than PF. To compare the signaling for periodic RSS and event-triggered proximity reports, we checked the total number of reports generated in both cases. There are 800 periodic RSS reports, while the number is significantly reduced to 74 for eventtriggered proximity reports. Even though not shown here, the effective sample size was on average 1513 for RSS and 1518 for proximity based particle filtering.

It is also 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 be "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 that 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 simulated measurements. More advanced modeling is studied in our work [25].







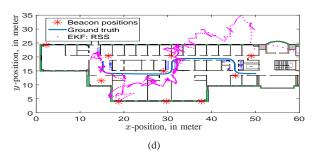


Fig. 9. Illustration of position estimates and ground truth with experimental data: 9(a) static MLE, 9(b) PF, 9(c) FFBSi, 9(d) EKF.

The overall CDF of the position errors for the same set of state trajectories in Fig. 8 are depicted in Fig. 10. The estimation errors at the 50-th, 67-th, and 95-th percentiles are listed in Table III. With experimental data, approximately 1 meter degradation in the positioning accuracy is observed for PF based on proximity report as compared to RSS at the 50-th percentile. By applying FFBSi smoothing, the positioning performance for proximity report is improved. The effects of different RSS thresholds are studied for experimental data and results are shown in Fig. 11. Lower  $\overline{\text{RCMSE}}$  is achieved at  $P_{\text{th}} = -82 \text{ dBm}$ . This is reasonable since with a high threshold (e.g. -70 dBm), the receiver may not be within any beacon's proximity, while a low threshold may lead to the case that the receiver is within all beacon's proximity. In both cases, little

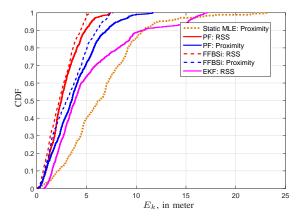


Fig. 10. CDF of position errors with experimental data.

information can be obtained, and the positioning accuracy may degrade due to the loss of information.

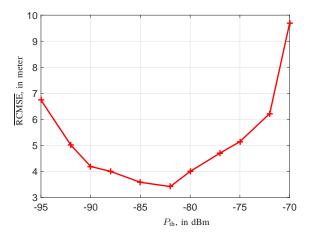


Fig. 11.  $\overline{\text{RCMSE}}$  versus different thresholds  $P_{\text{th}}$  with experimental data.

In what follows, we further evaluate the proposed method with different trajectories in the same building and a different sampling interval  $T_s$ . All the tested trajectories can be found in Section 3.1 of the supplementary report [30]. In the following experiments, we use the same particle filtering and smoothing parameters as listed in Table II, except that the sampling interval is set to  $T_s=1$  second. With an advertisement interval of around 100 ms, up to 10 samples can be measured within this interval. Then the averaged value of all these measurements are used to approximate the mean RSS value during this time interval. The averaging can be easily done either on the mobile device or at the computation entity by applying a low-pass filter within a time window of  $T_s$  second.

It is worth mentioning that the selection of sampling time  $T_s$  and how many samples can be averaged may depend on the advertisement interval of BLE beacons. For instance, the sampling interval is set to  $T_s=1$  second and up to 10 samples are averaged regarding that the BLE specification under this setup limits the shortest advertisement interval to 100 ms. However, considering the most recent BLE specification (i.e., Blue-tooth version 5.0), shorter advertisement interval (i.e.,

every 20ms) is designated. In such case, it is possible to average up to 50 samples within an interval of  $T_s=1$  second. With this specification, a shorter sampling interval can be selected (for instance, 0.5 second). It should also be noted that  $T_s$  cannot be too large, otherwise the position change within this interval would be too significant. Besides, with  $T_s$  larger than 1 second, the performance of particle smoother degrades greatly.

Due to space limitation, the position estimates for two representative trajectories are illustrated in Fig. 12. Similar performance are observed for other trajectories. The estimated trajectories are still satisfactory as the sampling interval is increased. The CDF of position errors for all tested trajectories are plotted in Fig. 13. From the CDF, the overall positioning accuracy across various trajectories is extracted in Table IV, which is similar to the previous case where a specific trajectory in Fig. 9 is used with shorter sampling interval. Besides, as we increase the sampling interval, the particle smoother performs worse than the particle filter. But the difference is not significant. However, by using particle smoother, smoothed trajectories can be obtained as can be seen from Fig. 12, which are usually preferred for post-processing and analyzing.

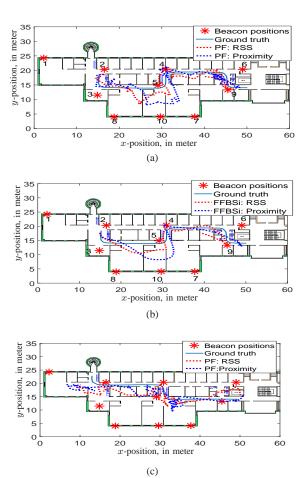
TABLE IV
POSITION ERRORS (IN UNIT OF METER) AT DIFFERENT PERCENTILES FOR
VARIOUS TRAJECTORIES.

Percentile		50-th	67-th	95-th
Experimental	PF(RSS/Proximity)	2.8/3.5	3.8/4.5	6.7/7.1
	FFBSi(RSS/Proximity)	3.1/3.8	4.1/4.9	6.7/8.1

#### VII. CONCLUSIONS

In this paper, we have proposed particle filtering and smoothing algorithms which are applicable for indoor positioning based on time series proximity reports. Such a proximity report based positioning algorithm is beneficial in different aspects, among which the signaling overhead can be significantly reduced via sending binary proximity measurements (i.e. 1 bit) approximately 10 times less frequently, in stead of e.g., periodic RSS values. The positioning performance based on proximity report is comparable with the performance based on periodic RSS reports. By applying particle smoothing, the positioning accuracy is improved for both simulated and experimental data. Furthermore, the positioning accuracy is close to the parametric CRBs. With experimental data, it can be concluded that 3 to 4 meter accuracy is achievable for most cases in this indoor scenario.

The work in this paper also motivates new research directions. For instance, the BLE locations are assumed to be known and static in this work. However, it would be interesting to see how much variations we can get for the same reference network node but at various locations. The training of propagation parameters can also be improved by introducing some on-line estimation methods, such as on-line expectation-maximization (EM) algorithm in [51] and [52]. Also, in order to improve the positioning accuracy, more complex propagation model may be explored in the near future. Besides, there are some other factors that can further



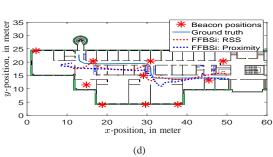


Fig. 12. Illustration of position estimates and ground truth with experimental data: 12(a) Trajectory 1, PF, 12(b) Trajectory 1, FFBSi, 12(c) Trajectory 2, PF, 12(d) Trajectory 2, FFBSi.

assist positioning, which includes but not limits to, the floor plan, inertial measurement unit (IMU) sensor measurements and so on. It would also be interesting to look into the effects of beacon placement and density.

We are currently working on Gaussian processes with multiple linear kernel based fundamental filtering and smoothing bounds, primarily based on the most recent results given in [53]–[56].

#### **APPENDIX**

For computing the parametric CRB of filtering and smoothing, it is required to further specify  $H_k(\boldsymbol{x}_k^*)^T R_k^{-1} H_k(\boldsymbol{x}_k^*)$  for

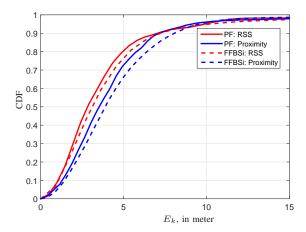


Fig. 13. CDF of position errors for all trajectories.

proximity and RSS measurements. According to [22], we can generally write

$$H_{k}(\boldsymbol{x}_{k}^{*})^{T} R_{k}^{-1} H_{k}(\boldsymbol{x}_{k}^{*})$$

$$= \mathbb{E} \left\{ -\Delta_{\boldsymbol{x}_{k}}^{\boldsymbol{x}_{k}} \ln p(\boldsymbol{y}_{k} | \boldsymbol{x}_{k}) \right\} \Big|_{\boldsymbol{x}_{k} = \boldsymbol{x}_{k}^{*}}$$

$$= \mathbb{E} \left\{ \left[ \nabla_{\boldsymbol{x}_{k}} \ln p(\boldsymbol{y}_{k} | \boldsymbol{x}_{k}) \right] [\cdot]^{T} \right\} \Big|_{\boldsymbol{x}_{k} = \boldsymbol{x}_{k}^{*}}.$$
(24)

In case of having RSS measurements, it is easy to verify that the corresponding expression simplifies to

$$\mathbb{E}\left\{-\Delta_{\boldsymbol{x}_{k}}^{\boldsymbol{x}_{k}} \ln p(\boldsymbol{y}_{R,k}|\boldsymbol{x}_{k})\right\}\Big|_{\boldsymbol{x}_{k}=\boldsymbol{x}_{k}^{*}}$$

$$= H_{k}(\boldsymbol{x}_{k}^{*})^{T} R_{k}^{-1} H_{k}(\boldsymbol{x}_{k}^{*}), \quad (26)$$

with

$$H_k(\boldsymbol{x}_k^*) = \nabla_{\boldsymbol{x}_k} \boldsymbol{h}^T(\boldsymbol{x}_k) \Big|_{\boldsymbol{x}_k = \boldsymbol{x}_k^*}, \tag{27a}$$

$$R_k = \operatorname{diag}([\sigma_1^2, \dots, \sigma_{N_B}^2]). \tag{27b}$$

The expressions for proximity measurements are a bit more involved. We introduce  $y_{P,k}^*$  which is a realization of the proximity measurement vector  $y_{P,k}$  and define the set S as all possible proximity measurements combinations  $2^{N_B}$ . Thus, we can write

$$H_{k}(\boldsymbol{x}_{k}^{*})^{T} R_{k}^{-1} H_{k}(\boldsymbol{x}_{k}^{*})$$

$$= \mathbb{E}\left\{ \left[ \nabla_{\boldsymbol{x}_{k}} \ln p(\boldsymbol{y}_{P,k} | \boldsymbol{x}_{k}) \right] [\cdot]^{T} \right\} \Big|_{\boldsymbol{x}_{k} = \boldsymbol{x}_{k}^{*}}$$

$$= \sum_{\boldsymbol{y}_{P,k}^{*} \in S} \left[ \nabla_{\boldsymbol{x}_{k}} \ln p(\boldsymbol{y}_{P,k}^{*} | \boldsymbol{x}_{k}) \right] [\cdot]^{T} p(\boldsymbol{y}_{P,k}^{*} | \boldsymbol{x}_{k}) \Big|_{\boldsymbol{x}_{k} = \boldsymbol{x}_{k}^{*}}$$
(28)

Further, due to the independence assumption, cf. (10), we can finally write

$$H_{k}(\boldsymbol{x}_{k}^{*})^{T} R_{k}^{-1} H_{k}(\boldsymbol{x}_{k}^{*})$$

$$= \sum_{y_{k,1}^{*} \in \{0,1\}} \cdots \sum_{y_{k,N_{B}}^{*} \in \{0,1\}} \left[ \sum_{j=1}^{N_{B}} \frac{\nabla_{\boldsymbol{x}_{k}} p(y_{k,j}^{*} | \boldsymbol{x}_{k})}{p(y_{k,j}^{*} | \boldsymbol{x}_{k})} \right] \left[ \cdot \right]^{T}$$

$$\times \prod_{j=1}^{N_{B}} p(y_{k,j}^{*} | \boldsymbol{x}_{k}) \bigg|_{\boldsymbol{x}_{k} = \boldsymbol{x}_{k}^{*}}, \tag{29}$$

where  $p(y_{k,i}^*|\boldsymbol{x}_k)$  is given in (11), and

$$\nabla_{\boldsymbol{x}_{k}} p(\boldsymbol{y}_{k,j}^{*} = 0 | \boldsymbol{x}_{k})$$

$$= \begin{bmatrix} -\frac{10B_{j}}{\ln 10\sqrt{2\pi}\sigma_{j}} \exp\left[-\frac{(P_{\text{th}} - \mu_{k,j})^{2}}{2\sigma_{j}^{2}}\right] \frac{(\bar{\mathbf{p}}_{x,k} - \mathbf{p}_{x,j})}{d_{k,j}^{2}} \\ 0 \\ -\frac{10B_{j}}{\ln 10\sqrt{2\pi}\sigma_{j}} \exp\left[-\frac{(P_{\text{th}} - \mu_{k,j})^{2}}{2\sigma_{j}^{2}}\right] \frac{(\bar{\mathbf{p}}_{y,k} - \mathbf{p}_{y,j})}{d_{k,j}^{2}} \\ 0 \end{bmatrix},$$
(30a)

$$\nabla_{x_k} p(y_{k,j}^* = 1 | x_k) = -\nabla_{x_k} p(y_{k,j}^* = 0 | x_k). \tag{30b}$$

#### ACKNOWLEDGMENT

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

Author Feng Yin is funded by the NSFC with the grant number 61701426 and the Shenzhen Science and Technology Innovation Council with the grant number JCYJ20170307155957688 and JCYJ20170411102101881.

#### REFERENCES

- [1] Y. Zhao, F. Yin, F. Gunnarsson, M. Amirijoo, E. Özkan, and F. Gustafsson, "Partile filtering for positioning based on proximity report," in *Proc. Int. Conf. on Information Fusion*, Washington D.C., USA, July 2015, pp. 1046–1052.
- [2] F. Gustafsson, Statistical Sensor Fusion, 2nd ed. Lund, Sweden Studentlitteratur, 2012.
- [3] "European Union FP7 Tracking in Complex Sensor Systems (TRAX) project," https://www.trax.utwente.nl/.
- [4] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response." ACM Comput. Surv., vol. 46, no. 2, p. 25, 2013.
- [5] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in Proc. 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Sept 2014, pp. 111– 120.
- [6] N. Patwari and A. O. Hero, III, "Using proximity and quantized RSS for sensor localization in wireless networks," in *Proc. ACM International Conference on Wireless Sensor Networks and Applications*. New York, NY, USA: ACM, 2003, pp. 20–29.
- [7] R. Karlsson and F. Gustafsson, "Filtering and estimation for quantized sensor information," in *Proc. 13th European Signal Processing Confer*ence (EUSIPCO), Antalya, Turkey, Sept. 2005.
- [8] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," ACM Trans. Inf. Syst., vol. 10, no. 1, pp. 91–102, Jan. 1992.
- [9] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.
- [10] 3GPP, "Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 25.331.
- [11] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.331.
- [12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proc. Annual International Conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2003, pp. 81–95.
- [13] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: a sequence based technique for RF localization in wireless sensor networks," in *Proc. Information Processing in Sensor Networks (IPSN)*, Los Angeles, California, USA, April 2005, pp. 285–292.
- [14] N. A. M. Maung and M. Kawai, "Performance improvement of connectivity-based localization using iterative learning," in *Proc. Int. Conf. Localization and GNSS (ICL-GNSS)*, Helsinki, Finland, June 2014, pp. 1–6.

- [15] R. Faragher and R. Harle, "An analysis of the accuracy of Bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute* of Navigation, Tampa, Florida, USA, September 2014, pp. 201–210.
- [16] J. Pelant, Z. Tlamsa, V. Benes, L. Polak, O. Kaller, L. Bolecek, J. Kufa, J. Sebesta, and T. Kratochvil, "BLE device indoor localization based on RSS fingerprinting mapped by propagation modes," in 27th International Conference Radioelektronika (RADIOELEKTRONIKA), April 2017, pp. 1–5.
- [17] F. Yin, Y. Zhao, and F. Gunnarsson, "Fundamental bounds on position estimation using proximity reports," in *Proc. IEEE 83rd Vehicular Technology Conference: VTC2016-Spring*, Nanjing, China, 2016, pp. 1–5.
- [18] N. Bergman, "Recursive Bayesian estimation navigation and tracking applications," Ph.D. dissertation, Linköping University, 1999.
- [19] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.
- [20] Z. Duan, V. P. Jilkov, and X. R. Li, "State estimation with quantized measurements: Approximate MMSE approach," in *Proc. 11th Interna*tional Conference on Information Fusion, June 2008, pp. 1–6.
- [21] L. Zuo, R. Niu, and P. K. Varshney, "A sensor selection approach for target tracking in sensor networks with quantized measurements," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, March 2008, pp. 2521–2524.
- [22] C. Fritsche, U. Orguner, and F. Gustafsson, "On parametric lower bounds for discrete-time filtering," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 4338–4342.
- [23] F. Yin, Y. Zhao, and F. Gunnarsson, "Proximity report triggering threshold optimization for network-based indoor positioning," in *Proc. Int. Conf. on Information Fusion*, Washington D.C., USA, July 2015, pp. 1061–1069.
- [24] K. P. Murphy, Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.
- [25] Y. Zhao, F. Yin, F. Gunnarsson, M. Amirijoo, and G. Hendeby, "Gaussian processes for propagation modeling and proximity based indoor positioning," in *Proc. IEEE 83rd Vehicular Technology Conference: VTC2016-Spring*, Nanjing, China, May 2016, pp. 1–5.
- [26] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," Wireless Communications & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, pp. 483–502, 2002.
- [27] N. Salman, N. Alsindi, L. Mihaylova, and A. H. Kemp, "Super resolution WiFi indoor localization and tracking," in *Proc. Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Oct 2014, pp. 1–5.
- [28] L. Bruno, M. Khider, and P. Robertson, "On-line training of the path-loss model in Bayesian WLAN indoor positioning," in *Proc. International Conference on Indoor Positioning and Indoor Navigation*, Oct 2013, pp. 1–9.
- [29] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li, Estimation with Applications to Tracking and Navigation. New York, NY, USA: John Wiley & Sons, Inc., 2002.
- [30] Y. Zhao, C. Fritsche, F. Yin, F. Gunnarsson, and F. Gustafsson, "Supplementary material for "Sequential Monte Carlo methods and theoretical bounds for proximity report based indoor positioning"," Linköping University, Department of Electrical Engineering, Linköping, Sweden, Tech. Rep. LiTH-ISY-R, 3101, 2017. [Online]. Available: http://www.control.isy.liu.se/publications/
- [31] M. Hata, "Empirical formula for propagation loss in land mobile radio services," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 3, pp. 317–325, Aug 1980.
- [32] A. Mousa, Y. Dama, M. Najjar, and B. Alsayeh, "Optimizing outdoor propagation model based on measurements for multiple RF cell," *International Journal of Computer Applications*, vol. 60, no. 5, pp. 5–10, December 2012.
- [33] A. Goldsmith, Wireless Communications. New York, NY, USA: Cambridge University Press, 2006.
- [34] S. Särkkä, Bayesian Filtering and Smoothing. Cambridge University Press., 2013.
- [35] R. E. Kalman, "A new approach to linear filtering and prediction problems," ASME Journal of Basic Engineering, 1960.
- [36] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," AIAA Journal, vol. 3, pp. 1445–1450, Aug. 1965.

- [37] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.
- [38] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Nonlinear Filtering Handbook*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2011, pp. 656–704.
- [39] J. D. Hol, T. B. Schön, and F. Gustafsson, "On resampling algorithms for particle filters," in *Proc. IEEE Nonlinear Statistical Signal Processing* Workshop, Sept. 2006, pp. 79–82.
- [40] M. Briers, A. Doucet, and S. Maskell, "Smoothing algorithms for state-space models," *Annals of the Institute of Statistical Mathematics*, vol. 62, no. 1, p. 61, 2009.
- [41] F. Lindsten, P. Bunch, S. Särkkä, T. B. Schön, and S. J. Godsill, "Rao-Blackwellized particle smoothers for conditionally linear Gaussian models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 353–365, March 2016.
- [42] C.-B. Chang and K.-P. Dunn, Applied State Estimation and Association. MIT press, 2016.
- [43] C. Fritsche and U. Orguner, "On parametric smoothing Cramér-Rao bounds," Linköping University, Department of Electrical Engineering, Linköping, Sweden, Tech. Rep. LiTH-ISY-R, 3097, 2017.
- [44] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "TOA-based robust wireless geolocation and Cramér-Rao lower bound analysis in harsh LOS/NLOS environments," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, pp. 2243–2255, May 2013.
- [45] I. Guvenc and C. C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.
- [46] G. Casella and R. L. Berger, Statistical inference, ser. Duxbury advanced series. Pacific Grove (Calif.): B. Cole, 2002.
- [47] E. S. Lohan, J. Talvitie, P. F. e Silva, H. Nurminen, S. Ali-Löytty, and R. Piché, "Received signal strength models for WLAN and BLEbased indoor positioning in multi-floor buildings," in *Proc. International Conference on Location and GNSS (ICL-GNSS)*, June 2015, pp. 1–6.
- [48] R. Khan, F. Sottile, and M. Spirito, "Hybrid positioning through extended Kalman filter with inertial data fusion," *International Journal of Information and Electronics Engineering*, Jan 2013.
- [49] J. Lategahn, M. Muller, and C. Rohrig, "TDoA and RSS based extended Kalman filter for indoor person localization," in *Proc. IEEE 78th Vehicular Technology Conference (VTC Fall)*, Sept 2013, pp. 1–5.
- [50] X. Shi, Y. H. Chew, C. Yuen, and Z. Yang, "A RSS-EKF localization method using HMM-based LOS/NLOS channel identification," in *Proc. IEEE International Conference on Communications (ICC)*, June 2014, pp. 160–165.
- [51] C. Andrieu, A. Doucet, and V. Tadic, "On-line parameter estimation in general state-space models," in *Proc. IEEE Conf. on Decision and Control*, Jan 2006.
- [52] Y. Li, X. Chen, M. Coates, and B. Yang, "Sequential Monte Carlo radio-frequency tomographic tracking," in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2011, pp. 3976–3979.
- [53] T. Chen, "Continuous-time DC kernel a stable generalized first-order spline kernel," *IEEE Transactions on Automatic Control, under review*, 2017
- [54] B. Mu and T. Chen, "On input design for regularized LTI system identification: Power-constrained inputs," *Automatica, under review*, 2017
- [55] T. Chen, "On kernel design for regularized LTI system identification," Automatica, to appear, 2017.
- [56] B. Mu, T. Chen, and L. Ljung, "On asymptotic properties of hyperparameter estimators for kernel-based regularization methods," *Automatica*, under review, 2017.