Parametric Lower Bound for Nonlinear Filtering based on Gaussian Process Regression Model

Yuxin Zhao, Carsten Fritsche and Fredrik Gunnarsson

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-140531

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

Original publication available at:
https://doi.org/10.23919/ICIF.2017.8009640
Copyright: IEEE
http://www.ieee.org/

©2017 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.
Parametric Lower Bound for Nonlinear Filtering based on Gaussian Process Regression Model

Yuxin Zhao*, Carsten Fritsche†, Fredrik Gunnarsson*

*Ericsson Research, Linköping, Sweden
†Division of Automatic Control, Linköping University, Linköping, Sweden
Email: {first name.last name}@ericsson.com, carsten@isy.liu.se

Abstract—Assessing the fundamental performance limitations in Bayesian filtering can be carried out using the parametric Cramér-Rao bound (CRB). The parametric CRB puts a lower bound on mean square error (MSE) matrix conditioned on a specific state trajectory realization. In this work, we derive the parametric CRB for state-space models, where the measurement equation is modeled by a Gaussian process regression. These models appear, for instance in proximity report-based positioning, where proximity reports are obtained by hard thresholding of received signal strength (RSS) measurements, that are modeled through Gaussian process regression. The proposed parametric CRB is evaluated on selected state trajectories and further compared with the positioning performance obtained by the particle filter. The results corroborate that the positioning accuracy achieved in this framework is close to the parametric CRB.

I. INTRODUCTION

A. Background

A full framework of proximity report based indoor positioning has been given in [1], where proximity is defined by hard-thresholding of a received signal strength (RSS) measurement. An illustration of generating proximity reports is shown in [1, Figure 1]. Unlike periodic RSS reports (which contain RSS measurement vectors), where a mobile device sends the measured RSS values to the computation entity for positioning purpose regularly, a proximity report (contains the proximity measurement vector) will be triggered only when the proximity status changes. A proximity report obtained in this way reveals whether or not the reporting device is in the proximity (depending on the threshold) of a reference network node, for instance a blue-tooth low energy (BLE) beacon, with known position. Since proximity reports are triggered occasionally, this will result in much less signaling overhead.

The positioning performance of such a system relies heavily on the accuracy of the measurement model. Hence, Gaussian process regression (GPR) model was motivated to present an RSS measurement in terms of the 2-D geographical position in a more realistic indoor environment. In [1], we have proposed a particle filter that models the RSS measurement using GPR. Experimental results show that with GPR model, the positioning performance can be improved due to the more accurate measurement model.

Assessing the theoretical performance limits of any estimator, given a state-space model with GPR measurement model, is investigated in this paper. The Cramér-Rao bound (CRB) is often applied as a benchmark to bound the covariance of any estimator from below. For Bayesian filtering, posterior and parametric CRBs are usually considered. The posterior CRB provides a lower bound on the mean squared error (MSE) of any state estimator $\hat{x}_k(y_{1:k})$, i.e.

$$\text{MSE}(\hat{x}_k(y_{1:k})) \triangleq \mathbb{E}\{(\hat{x}_k(y_{1:k}) - x_k)^T\} \succeq \text{PosCRB},$$

(1)

where $x_k$ denotes the state at time $k$ and $y_{1:k}$ are measurements from time 1 to $k$, and the operator $\succeq$ means that the difference $\text{MSE}(\hat{x}_k(y_{1:k})) - \text{PosCRB}$ is positive semi-definite.

Posterior CRBs have the limitation that generally no closed-form solutions are available, if the state-space model is nonlinear [2]. In such cases, one has to approximate the expectations appearing in the recursions by Monte Carlo averages which require the simulation of a set of state and measurement realizations. However, in indoor scenarios as it is considered in this work, it is not easy to generate state trajectories from a motion model that remain inside the building. Therefore, the parametric CRB is an interesting alternative, as it puts a bound on the MSE of any unbiased estimator $\hat{x}_k(y_{1:k})$, given a specific deterministic state trajectory $x_{0:k}$, i.e.

$$\mathbb{E}\{(\hat{x}_k(y_{1:k}) - x_k)^T \mid x_{0:k}\} \succeq \text{ParaCRB}(x_{0:k}).$$

(2)

As it has been shown in [3], the state trajectory to be selected inside the building can be arbitrary, and it not necessarily has to be generated from the motion model.

B. Related Work and Our Contributions

In [4], the parametric CRB has been calculated for discrete-time nonlinear filtering problems with the assumption that the deterministic state trajectory has been generated from a process model with zero process noise. In [3], it has been shown that this assumption can be relaxed and the parametric CRB also holds deterministic state trajectories that have been generated from a model with nonzero process noise. In our current ongoing work [5], the parametric filtering and smoothing CRBs for proximity report based positioning have been derived, where the RSS measurement is modeled by a simple linear log-distance relation.

However, as already discussed the more complex GPR model will better approximate RSS measurements, and thus will yield a different parametric CRB. To the best of our knowledge, the calculation of the parametric filtering CRB...
has never been derived for state-space models with GPR measurement models.

Our contributions of this paper are summarized as follows. Based on the proximity report based positioning framework using Gaussian processes [1], we derive the corresponding parametric filtering CRBs for both RSS and proximity report measurements modeled using GPR, which provide theoretical lower limits for the positioning performance under this framework.

C. Paper Organization and Notations

The remainder of this paper is organized as follows: Section II reviews in detail the system model the particle filtering algorithm based on proximity report. Section III gives a detailed derivation of the parametric filtering CRB for GPR model based nonlinear filtering problems. Section IV calculates and compares the derived parametric CRBs in various experiments. Lastly, Section V concludes the paper.

Throughout this paper, matrices are presented with uppercase letters and vectors with boldface lowercase letters. The operator $[·]^T$ stands for vector/matrix transpose and $[·]^{-1}$ stands for the inverse of a non-singular square matrix. $∥·∥$ denotes the Euclidean norm of a vector and $|·|$ denotes the cardinality of a set. The operator $\mathbb{E}(·)$ denotes the statistical expectation. $X(·)^T$ is a short-hand notation for $XX^T$. The operator $\ln(·)$ stands for the natural logarithm and $\log_{10}(·)$ stands for the logarithm to base 10. In addition, the operator $\otimes$ denotes the Kronecker product. Further, $\nabla_\theta = \partial/\partial \theta$ denotes the gradient operator and $\Delta_\theta = \nabla_\theta \nabla_\theta^T$ denotes the Laplace operator. $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean $\mu$ and variance $\sigma^2$. $\text{diag}(x)$ denotes the diagonal matrix with the elements of vector $x$ on the main diagonal.

II. GAUSSIAN PROCESSES FOR PROXIMITY REPORT BASED PARTICLE FILTERING

A. System Model

We consider an indoor sensor network that consists of $N_B$ reference network nodes with a priori known positions, $p_{j,1}, j = 1, 2, \ldots, N_B$ and one computation entity. A set of model related parameters are calibrated during a training phase. In an operational phase the state of a mobile device, $x_k$, is to be estimated at each time step $k$. The state-space model used in this work is given by:

$$x_{k+1} = Fx_k + w_k,$$

$$y_k = h_k(x_k, e_k),$$

where (3a) is the linear motion model as given in [6, Chapter 6] and (3b) is the measurement model. Here $w_k$ is the process noise, $e_k$ is the measurement noise, $y_k$ denotes the measurement vector, and $h_k(·)$ is a nonlinear function of appropriate size.

**Motion model:** The state vector $x_k = [p_{x,k}, p_{y,k}, v_{x,k}, v_{y,k}]^T$ consists of 2-D position vector $p_k = [p_{x,k}, p_{y,k}]$ and velocity $v_k = [v_{x,k}, v_{y,k}]$, respectively. The transition matrix $F$ is given by

$$F = I_2 \otimes \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix},$$

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

$$\Sigma_w = I_2 \otimes \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix},$$

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

**Measurement model:** The measurement model relates the measurements to the unknown state. In the sequel, we give the RSS and proximity measurement models based on GPR. From [1], we can represent an RSS measurement from reference network node $j$ in terms of a Gaussian process at time $k$, which is given by

$$r_{k,j}(p_k) \sim \mathcal{GP}(m_j(p_k), k_j(p_k, p_k')) + e_{k,j},$$

with

$$m_j(p_k) \equiv A_j + 10 B_j \log_{10} \left( \frac{∥p_k - p_{j,s}∥}{d_0} \right),$$

$$k_j(p_k, p_k') \equiv \sigma_{s,j}^2 \cdot \exp \left[ -\frac{∥p_k - p_k'∥}{l_{s,j}} \right].$$

where $A_j$ is the received power at reference distance $d_0$, $B_j$ is the path loss component, $e_{k,j}$ is assumed to be time invariant i.i.d. Gaussian noise with zero mean and variance $\sigma_{e,j}^2$, $l_{s,j}$ is the correlation distance and $\sigma_{s,j}^2$ reflects the variance of the above Gaussian kernel function in (5b) [7].

Hence, the RSS measurement vector consists of measurements from all nodes at time $k$ can be given by

$$y_{rk,k} = [r_{k,1}, \ldots, r_{k,N_B}]^T.$$

Correspondingly, we have the proximity measurement vector

$$y_{pr,k} = y_{pr,k} = [y_{k,1}, \ldots, y_{k,N_B}]^T, \quad g(y_{rk,k}) = |g(r_{k,1}, \ldots, g(r_{k,N_B})|^T$$

and $g(x)$ is a non-linear function which performs hard thresholding of an input, $x$, as follows:

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

For the $j$-th given reference network node, define a set of training data as $T_j = \{\{\bar{p}_j, r_j(\bar{p}_j')\}| s = 1, 2, \ldots, S_j\}$, the joint distribution of all observed RSS measurements $\{r_j(\bar{p}_j)| s = 1, 2, \ldots, S_j\}$ can be given as

$$p(r_j(P_j); \theta_j) \sim \mathcal{N}(m_j(P_j), C(P_j, P_j)),$$
with the following definitions

\[ \theta_j = [A_j, B_j, \sigma_{\epsilon_j}^2, \epsilon_j], \]
\[ P_j \triangleq [p_j, p_j^2, \ldots, p_j^S_j], \]
\[ r_j(P_j) \triangleq [r_1(p_j), r_2(p_j^2), \ldots, r_S_j(p_j^S_j)]^T, \]
\[ m_j(P_j) \triangleq [m_1(p_j), m_2(p_j^2), \ldots, m_S_j(p_j^S_j)]^T, \]
\[ K(P_j, P_j) \triangleq \begin{bmatrix} k_j(p_j, p_j^1) & k_j(p_j, p_j^2) & \ldots & k_j(p_j, p_j^S_j) \\ k_j(p_j^1, p_j) & k_j(p_j^1, p_j^2) & \ldots & k_j(p_j^1, p_j^S_j) \\ \vdots & \vdots & \ddots & \vdots \\ k_j(p_j^S_j, p_j^1) & k_j(p_j^S_j, p_j^2) & \ldots & k_j(p_j^S_j, p_j^S_j) \end{bmatrix}, \]
\[ C(P_j, P_j) \triangleq K(P_j, P_j) + \sigma_{\epsilon_j}^2 I_{S_j}. \]

The parameters included in \( \theta_j \) are usually unknown and need to be calibrated. A parameter estimate, \( \hat{\theta_j} \), can be found through maximizing the likelihood function (6) numerically using for instance the limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) quasi-Newton method or the conjugate gradient (CG) method [8]. The obtained \( \hat{\theta_j} \) is treated then as underlying parameters.

In order to give a training data dependent observed RSS model, we compute according to [7] the Gaussian posterior probability of an observed RSS value at any position \( p_k \) from \( j \)-th reference node by

\[ p(r_j(p_k)|\theta_j) \sim N(\overline{\psi}_j(p_k), \overline{\sigma}_j^2(p_k)), \] (7)

where

\[ \overline{\psi}_j(p_k) = k_j^T(p_k, P_j)C^{-1}(P_j, P_j)(r_j(P_j) - m_j(P_j)) + m_j(p_k), \] (8a)
\[ \overline{\sigma}_j^2(p_k) = \sigma_{\epsilon_j}^2 + k_j^2(p_k, P_j)C^{-1}(P_j, P_j)k_j(p_k, P_j). \] (8b)

Note that in (8a) and (8b),

\[ k_j(p_k, P_j) \triangleq [k_j(p_k, p_j^1), k_j(p_k, p_j^2), \ldots, k_j(p_k, p_j^S_j)]^T. \]

**B. Particle Filtering for GPR based Models**

In this paper, the bootstrap particle filter in [9] is used. The importance density is chosen to be the transitional prior \( p(x_k|x_{k-1}) = N(x_k; Fx_{k-1}, \Sigma_u) \). The general particle filtering procedure is summarized in Algorithm 1.

As given in [1], the likelihood function for proximity measurement is evaluated as follows:

\[ p(y_{P,k}|x_k) = \prod_{j=1}^{N_p} p(y_{k,j}|x_k) \]
\[ = \prod_{j=1}^{N_p} \sum_{l \in \{0,1\}} \delta(y_{k,j} - l)p(y_{k,j} = l|x_k), \] (9)

where \( \delta(\cdot) \) denotes the Dirac delta function. We further have

\[ p(y_{k,j} = 0|x_k) = \Phi \left( \frac{P_k - \overline{\psi}_j(p_k)}{\overline{\sigma}_j(p_k)} \right), \] (10a)
\[ p(y_{k,j} = 1|x_k) = 1 - p(y_{k,j} = 0|x_k), \] (10b)

**Algorithm 1 Bootstrap Particle Filter**

1. **Initialization**: Draw \( N \) samples \( x_0^{(i)} \sim p(x_0) \), and set \( w_0^{(i)} = 1/N \), for all \( i = 1, \ldots, N \).
2. For \( k = 1, \ldots, T \), do:
   a. Draw \( N \) samples \( x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)}) \), for \( i = 1, \ldots, N \).
   b. Update weights according to
      \[ \tilde{w}_k^{(i)} \sim w_{k-1}^{(i)} p(y_k|x_k^{(i)}), \]
      for \( i = 1, \ldots, N \).
   c. For \( i = 1, \ldots, N \), normalize the weights to sum to unity,
      \[ w_{k}^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N} \tilde{w}_k^{(i)}}. \]
   d. An approximation to the posterior filtering expectation is given by
      \[ \hat{x}_{PF,k}(y_{1:k}) = \sum_{i=1}^{N} w_k^{(i)} x_k^{(i)} \approx E\{x_k|y_{1:k}\}. \]
   e. **Resampling**: If \( N_{eff} = 1/\sum_{i=1}^{N} (w_k^{(i)})^2 < N_{th} \), then perform multinomial resampling [10].

with \( \overline{\psi}_j(p_k) \) and \( \overline{\sigma}_j(p_k) \) given in (8a) and (8b), respectively and

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

For RSS measurements, the likelihood function is calculated as

\[ p(y_{R,k}|x_k) = \prod_{j=1}^{N_R} p(r_{k,j}|x_k) \]
\[ = \prod_{j=1}^{N_R} \frac{1}{\sqrt{2\pi\sigma_{\epsilon_j}(p_k)}} \exp \left( -\frac{(r_{k,j} - \overline{\psi}_j(p_k))^2}{2\sigma_{\epsilon_j}^2(p_k)} \right). \] (11)

**III. PARAMETRIC FILTERING CRB WITH GAUSSIAN PROCESS REGRESSION MODEL**

The parametric filtering CRB \( P_k(x_0^{*}) \) gives a lower bound on the conditional MSE (CMSE) matrix for any unbiased estimator \( \hat{x}_k(y_{1:k}) \) as follows

\[ \text{CMSE}(\hat{x}_k(y_{1:k}) | x_k^{*}) = E\{(\hat{x}_k(y_{1:k}) - x_k^{*})^T | x_k^{*}\} \geq P_k(x_k^{*}). \] (12)
model. The procedure for calculating the parametric CRB is given in Algorithm 2.

The key to compute the parametric filtering CRB with

Algorithm 2 Parametric Filtering CRB

1) Assuming the prior distribution is Gaussian \( p(x_0) = N(\mu_{x_0}, \Sigma_{x_0}) \), initialize the FIM with \( J_0(x_0) = \Sigma_{x_0}^{-1} \). The parametric filtering CRB is computed as \( P_0(x_0) = J_0^{-1}(x_0) \).
2) For \( k = 1, \ldots, T \), do:
   - Compute the filtering FIM from
     \[
     J_k = [F J_{k-1}^{-1} F^T]^{-1} + Q_k(x_k^*),
     \]
     where \( J_k \) is a short-hand notation for \( J_k(x_k^*) \), and
     \[
     Q_k(x_k^*) = \mathbb{E} \left\{ -\Delta x_k \ln p(y_k | x_k) \right\} |_{x_k^*},
     \]
   - Compute the parametric filtering CRB as
     \[
     P_k(x_k^*) = J_k^{-1}(x_k^*).
     \]

GPR model is to derive the term \( Q_k(x_k^*) \) according to the measurement model given in Section II-A. Derivations are given for RSS and proximity measurement in the sequel, respectively.

**I: RSS measurement based on GPR.** For RSS measurements represented by Gaussian processes, we can have

\[
y_{R,k} = [r_{k,1}, \ldots, r_{k,N_{th}}]^T = [\overline{\mu}_1(p_k) + \overline{\sigma}_1, \ldots, \overline{\mu}_{N_{th}}(p_k) + \overline{\sigma}_{N_{th}}]^T,
\]

where \( \overline{\sigma}_j \) is i.i.d Gaussian noise with zero mean and variance \( \overline{\sigma}_j^2(p_k) \), \( \overline{\mu}_j(p_k) \) and \( \overline{\sigma}_j(p_k) \) are posterior mean and variance of the Gaussian process which are given in (8a) and (8b), respectively. As the noise is additive and Gaussian distributed, it is easy to derive for RSS measurements based on GPR

\[
Q_k(x_k^*) = \mathbb{E} \left\{ -\Delta x_k \ln p(y_{R,k} | x_k) \right\} |_{x_k^*} = H_k(x_k^*)^T R_k^{-1} H_k(x_k^*),
\]

with

\[
H_k(x_k^*) = \nabla_{x_k} \overline{\mu}(p_k)^T |_{x_k^*}, \quad (14a)
\]

\[
R_k = \Sigma_{\sigma}.
\]

where

\[
\overline{\mu}(p_k) = [\overline{\mu}_1(p_k), \ldots, \overline{\mu}_{N_{th}}(p_k)]^T,
\]

and \( \Sigma_{\sigma} = diag(\overline{\sigma}_1^2(p_k), \ldots, \overline{\sigma}_{N_{th}}^2(p_k))^T \). Detailed derivations for computing \( \nabla_{x_k} \overline{\mu}(p_k) \) are given in (21) in the Appendix.

**II: Proximity measurement based on GPR.** For proximity measurement obtained by hard thresholding RSS values, the measurement noise is no longer additive. Hence, we derive the term \( Q_k(x_k^*) \) as follows

\[
Q_k(x_k^*) = \mathbb{E} \left\{ -\Delta x_k \ln p(y_{P,k} | x_k) \right\} |_{x_k^*}
\]

where \( y_{P,k} \) is a realization of the proximity measurement vector \( y_{P,k} \) and the set \( D \) contains all possible proximity measurements combinations \( 2^{N_{th}} \). Further, due to the independence assumption, cf. (9), we can finally write

\[
Q_k(x_k^*) = \sum_{y_{P,k} \in D} \left[ \nabla_{x_k} \ln p(y_{P,k} | x_k) \right] [y_{P,k} | x_k]^T p(y_{P,k} | x_k) |_{x_k^*},
\]

where \( p(y_{P,k} | x_k) \) is given in (10), and

\[
\nabla_{x_k} p(y_{P,k} | x_k) = 0 \Rightarrow \nabla_{x_k} \Phi \left( \frac{P_{th} - \overline{\mu}_j(p_k)}{\overline{\sigma}_j(p_k)} \right), \quad (16a)
\]

\[
\nabla_{x_k} p(y_{P,k} | x_k) = 1 \Rightarrow -\nabla_{x_k} \Phi \left( \frac{P_{th} - \overline{\mu}_j(p_k)}{\overline{\sigma}_j(p_k)} \right), \quad (16b)
\]

with \( \overline{\mu}_j(p_k) \) and \( \overline{\sigma}_j(p_k) \) are the posterior mean and variance of the Gaussian process which are given in (8a) and (8b), respectively.

Now the main task is to derive \( \nabla_{x_k} \Phi \left( \frac{P_{th} - \overline{\mu}_j(p_k)}{\overline{\sigma}_j(p_k)} \right) \), which is given in (25) in the Appendix.

**IV. Results**

**A. Measurement Campaign**

We consider a typical office environment at Ericsson, Linköping, Sweden. In total \( N_{th} = 10 \) 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, where, a local coordinate system is used. 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) receives data packages from the BLE beacons and measures the RSS. A total of \( M = 12144 \) RSS measurements were collected offline along 52 predefined tracks, from which \( \{ \theta_j \}_{j=1}^{N_{th}} \) are estimated. Then, optimized threshold \( P_{th} \) is set according to [11].
B. Performance Evaluation

After all model parameters $\{\hat{\theta}_j\}_{j=1}^{N_B}$ are estimated, RSS and proximity reports are simulated along selected state trajectories using the GPR model given in Section II-A. One example of selected trajectory is shown in Fig. 1. The state trajectory was generated for 280 time steps with a sampling interval of $T_s = 0.1$ sec. Then, combining the motion model and simulated measurements, particle filtering has been applied to obtain position estimates. The parameters specified for the particle filter are listed in Table I. $\sigma_w$ is a tuning parameter. Various values were tested with the particle filtering algorithm and $\sigma_w = 1$ gives good performance.

![Sensor deployment (with BLE beacons marked by red *) and one example of selected trajectory.](image)

**Fig. 1.** Sensor deployment (with BLE beacons marked by red *) and one example of selected trajectory.

**TABLE I**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_w$</td>
<td>1</td>
<td>Noise power spectral density</td>
</tr>
<tr>
<td>$T_s$</td>
<td>0.1 sec</td>
<td>Sampling interval</td>
</tr>
<tr>
<td>$P_{th}$</td>
<td>-84 dBm</td>
<td>Predefined RSS threshold</td>
</tr>
<tr>
<td>$N$</td>
<td>2000</td>
<td>Number of particles</td>
</tr>
<tr>
<td>$N_B$</td>
<td>10</td>
<td>Number of deployed reference nodes</td>
</tr>
<tr>
<td>$\hat{x}_{0}$</td>
<td>$[15, 1, 10, 10]^T$</td>
<td>Mean of $x_0$</td>
</tr>
<tr>
<td>$\Sigma_{x_0}$</td>
<td>diag([1, 2, 1])</td>
<td>Covariance of $x_0$</td>
</tr>
<tr>
<td>$\sigma_w$</td>
<td>$\frac{2}{3} N$</td>
<td>Threshold for resampling</td>
</tr>
</tbody>
</table>

In order to compare the performance of the particle filter with theoretical limits, we need to compare the two matrices given in (12), which is usually difficult in practice. Instead, consider the scalar root conditional MSE (RCMSE), giving the following relation for any conditionally unbiased estimator

$$\text{RCMSE}_k \geq \sqrt{P_{1,1}^{k|k} + P_{1,3}^{k|k} \triangleq \gamma_k}, \quad (17)$$

where

$$\text{RCMSE}_k \approx \sqrt{\frac{1}{M} \sum_{m=1}^{M} ||\hat{p}_k^{(m)} - \bar{p}_k||^2}. \quad (18)$$

Here $\hat{p}_k^{(m)}$ is the position estimate of the estimator $\hat{x}_k(y_{1:k}^{(m)})$ obtained from the $m$-th Monte Carlo run, $M$ is set to 1000 in this work, $\bar{p}_k$ denotes the ground truth, and $P^{i,j}$ denotes the element at row $i$ column $j$ of the parametric filtering CRB matrix $P_k(x_*^k)$. It should be noted that the relation in (17) only holds for conditionally unbiased estimator. However, in practical cases, bias is usually observed and the bias conditioned on a specific state trajectory is defined as

$$b_k \approx \frac{1}{M} \sum_{m=1}^{M} \left[ \hat{p}_k^{(m)} - \bar{p}_k \right]. \quad (19)$$

The RCMSE for state trajectory shown in Fig. 1 is compared with the parametric filtering CRBs in Fig. 2. It is observed that the estimate position error appears above the parametric filtering bound. It is also noticed that the positioning algorithm achieves RCMSE$_k$ that is close to the bound for most of the positions. By observing the bias shown in Fig. 3, it is noted that the bias in both x and y-axes is high for $k$ between 20 and 30 and bias in y-axes is high for $k$ between 100 and 150, which lead to larger gaps between the CRBs and RCMSE$_k$ at those occasions.

![RCMSE$_k$ and parametric filtering CRB.](image)

**Fig. 2.** RCMSE$_k$ and parametric filtering CRB.

![Bias in position estimates.](image)

**Fig. 3.** Bias in position estimates.
Based on GPR, we need to further derive

\[ f_i \]

linear filtering based on Gaussian process regression model. It is observed that the overall positioning accuracy is around 2 meters with proximity measurements and 1.5 meters with RSS measurements.

V. CONCLUSION

In this work, the parametric CRB has been derived for non-linear filtering based on Gaussian process regression model. The performance achieved by particle filtering is further compared with the theoretical lower bound. The positioning performance based on Gaussian process regression model is concluded to be close to the theoretical lower limits and the overall positioning accuracy in this office environment is around 2 meters with proximity measurements and 1.5 meters with RSS measurements.

APPENDIX

To assist the calculation of parametric filtering CRB based on GPR, we need to further derive \( \nabla_{x_k} \pi_j(p_k) \) and \( \nabla_{x_k} \Phi \left( \frac{F_k - \bar{p}_j(p_k)}{\tau_j(p_k)} \right) \).

\( \nabla_{x_k} \pi_j(p_k) \) can be further derived as

\[
\nabla_{x_k} \pi_j(p_k) = [\nabla_{x_k} k(p_k, P_j)] C(P_j, P_j) [r_j(P_j) - m_j(P_j)] + \nabla_{x_k} m_j(p_k),
\]

(21)

where \( \nabla_{x_k} m_j(p_k) \) can be easily obtained and for \( x_k = [p_{x,k}, v_{x,k}, p_{y,k}, v_{y,k}]^T \), we have

\[
\nabla_{x_k} k(p_k, P_j) = \begin{bmatrix}
\nabla_{p_{x,k}} k(p_k, P_j) \\
\nabla_{v_{x,k}} k(p_k, P_j) \\
\nabla_{p_{y,k}} k(p_k, P_j) \\
\nabla_{v_{y,k}} k(p_k, P_j)
\end{bmatrix},
\]

(22)

with the following equations hold

\[
\nabla_{p_{x,k}} k(p_k, P_j) = \begin{bmatrix}
\nabla_{p_{x,k}} k(p_k, P_j) \\
\nabla_{v_{x,k}} k(p_k, P_j) \\
\nabla_{p_{y,k}} k(p_k, P_j) \\
\nabla_{v_{y,k}} k(p_k, P_j)
\end{bmatrix},
\]

where the following equations hold

\[
\nabla_{p_{x,k}} k(p_k, P_j) = \begin{bmatrix}
\nabla_{p_{x,k}} k(p_k, P_j) \\
\nabla_{v_{x,k}} k(p_k, P_j) \\
\nabla_{p_{y,k}} k(p_k, P_j) \\
\nabla_{v_{y,k}} k(p_k, P_j)
\end{bmatrix},
\]

(24a)

\[
\nabla_{p_{y,k}} k(p_k, P_j) = \begin{bmatrix}
\nabla_{p_{x,k}} k(p_k, P_j) \\
\nabla_{v_{x,k}} k(p_k, P_j) \\
\nabla_{p_{y,k}} k(p_k, P_j) \\
\nabla_{v_{y,k}} k(p_k, P_j)
\end{bmatrix},
\]

(24b)

The derivative of \( \Phi \left( \frac{F_k - \bar{p}_j(p_k)}{\tau_j(p_k)} \right) \) with respect to state vector \( x_k \) can be computed as

\[
\nabla_{x_k} \Phi \left( \frac{F_k - \bar{p}_j(p_k)}{\tau_j(p_k)} \right) = \frac{1}{1 + \gamma_k} \nabla_{x_k} \pi_j(p_k) + \frac{1}{1 + \gamma_k} \frac{F_k - \bar{p}_j(p_k)}{\tau_j(p_k)} \nabla_{x_k} \tau_j(p_k),
\]

(25)

where \( \nabla_{x_k} \pi_j(p_k) \) is given in (21),

\[
\nabla_{x_k} \tau_j(p_k) = \frac{1}{\sqrt{2\pi\tau^2_j(p_k)}} \exp \left( \frac{(p_k - \bar{p}_j(p_k))^2}{2\tau^2_j(p_k)} \right),
\]

and

\[
\nabla_{x_k} \tau_j^2(p_k) = - (\nabla_{x_k} k(x_k, p_j) \left[ C_j^{-1} + (C_j^{-1})^T \right] k(x_k, P_j)),
\]

with \( C_j \) is the short-hand notation for \( C(P_j, P_j) \) and \( \nabla_{x_k} k(x_k, P_j) \) given by (22).

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. We acknowledge the support from Senion, who provided the BLE beacons and associated positioned RSS data.

REFERENCES


