# Cramér–Rao Bounds for Filtering Based on Gaussian Process State-Space Models

Yuxin Zhao, Carsten Fritsche, Gustaf Hendeby, Feng Yin, Tianshi Chen and Fredrik Gunnarsson

# Cramér-Rao Bounds for Filtering Based on Gaussian Process State-Space Models

Yuxin Zhao, *Member, IEEE,* Carsten Fritsche, *Member, IEEE,* Gustaf Hendeby, *Senior Member, IEEE,*
Feng Yin, *Member, IEEE,* Tianshi Chen, *Member, IEEE,* Fredrik Gunnarsson, *Senior Member, IEEE.*

***Abstract*—Posterior Cramér-Rao bounds (CRBs) are derived for the estimation performance of three Gaussian process-based state-space models. The parametric CRB is derived for the case with a parametric state transition and a Gaussian process-based measurement model. We illustrate the theory with a target tracking example and derive both parametric and posterior filtering CRBs for this specific application. Finally, the theory is illustrated with a positioning problem, with experimental data from an office environment where the obtained estimation performance is compared to the derived CRBs.**

***Index Terms*—Cramér-Rao bound, Gaussian process, state-space model, nonlinear estimation**

## I. Introduction

### A. Motivations and Background

The study of Gaussian process (GP)-based state-space models has been emerging in recent years. Dynamic models for the state formulated as Gaussian processes are introduced in [2]. There are also studies on Gaussian process-based measurement models, such as [3] and [4]. A full state-space model given by Gaussian process is provided in [5]. State estimation based on Gaussian process models have been investigated from various perspectives. For instance, inference and learning for Gaussian process state-space model is studied in [6], and particle filtering is applied to obtain state estimation sequentially for a Gaussian process-based measurement model in [7] and [8].

However, so far, there are barely any studies on the theoretical bounds for the above mentioned estimation problems formulated by Gaussian process state-space models, while understanding the performance limitation is essential from various aspects. For instance, assessing the theoretical lower limits can help to evaluate the performance of a certain estimation algorithm by measuring how far away the achieved performance is from the bound. Besides, since the theoretical bound provides a best achievable estimation performance under certain conditions, before designing any algorithm, the bound can be computed to have a general understanding about the estimation problem which needs to be solved.

For static estimation problems with Gaussian process models, different lower bounds have been derived, such as the Cramér-Rao Bounds (CRBs) given in [9] and [10]; Barankin bounds are derived in [11]. For dynamic estimation problems, which are usually formulated in a state-space form and solved sequentially, to our best knowledge, there is rarely any work touching upon the theoretical lower limits for state-space models formulated by Gaussian processes. To this end, we are aiming at deriving the theoretical lower bounds for Gaussian process state-space models. To be more specific, we focus on parametric and posterior filtering Cramér-Rao Bounds for Gaussian process state-space models in various forms.

### B. Related Work and Contributions

A dynamic estimation problem is usually formulated using a state-space representation and consists of two parts. The first part is the state transition model, which accounts for the evolution of the states over time. The second part is the measurement model, which describes how the outputs/measurements are related to the states. The aim is to estimate the unknown states from the measurements. There are also cases where the unknown parameters in the state transition and measurement model need to be estimated. One way to estimate the unknown parameters is to estimate them at the same time as sequentially estimating the states, see for instance [5]. Correspondingly, the theoretical lower limits on both the unknown states and parameters in the models can be derived, and the limits are known as hybrid bounds, see for instance [12]. Another way is to assume an off-line phase, where the unknown parameters are estimated from a set of pre-collected training data. Then, the states are estimated based on the models with known parameters. According to [13], the CRBs derived with deterministic parameters are generally tighter than the hybrid CRBs. In this work, we focus on the lower limits for states estimation rather than for the model parameters. Hence, the model parameters are considered as deterministically known.

A common choice for the state transition and measurement model is the parametric way. For instance, in a linear state-space model, both the state transition and measurement are given as linear functions [14]. However, both state transition and measurement can also be formulated in a non-parametric way, for instance, using a Gaussian process which imposes a Gaussian prior on the underlying function. Considering

This work is an extension of our conference paper [1].

Feng Yin is the correspondence author.

Y. Zhao was with the Department of Electrical Engineering, Division of Automatic Control, Linköping University, Linköping, SE-58183, Sweden from 2017 to 2019. Y. Zhao and F. Gunnarsson are with Ericsson Research, Linköping, SE-58330, Sweden. (E-mail: yuxin.zhao,fredrik.gunnarsson@ericsson.com).

C. Fritsche and G. Hendeby are with the Department of Electrical Engineering, Division of Automatic Control, Linköping University, Linköping, SE-58183, Sweden. (E-mail: carsten.fritsche,gustaf.hendeby@liu.se).

F. Yin and T. Chen are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, CN-518172, China. (E-mail: yinfeng/tschen@cuhk.edu.cn).

TABLE I: Four cases of state-space models

| Case | Description |
|---|---|
| 0 | Parametric state transition model and parametric measurement model |
| 1 | Parametric state transition model and non-parametric measurement model |
| 2 | Non-parametric state transition model and parametric measurement model |
| 3 | Non-parametric state transition model and non-parametric measurement model |

complex scenarios where it is very difficult to parametrize both state transition and measurement model, the complex parametric models are at the expense of increasing the number of parameters to be estimated. However, for complex modeling problems, a non-parametric approximation might be better, since the models are purely constructed from the off-line training data. This is especially beneficial when the training data set is big and the off-line computation power is not a big issue. It is good to mention that if either of the motion or measurement model can be represented precisely by a deterministic parametric model, there is no need to use non-parametric methods where usually a training phase is required to collect lager sets of training data. A more detailed introduction to Gaussian process regression as a non-parametric method will be provided in Section II. Hence, considering the fact that the state transition and measurement function can be in either parametric or non-parametric form, four different cases can be formulated, which are summarized in Table I. Since case 0 is the normal case which can be handled by standard methods, such as different types of Kalman filters, it is not studied in this work.

Gaussian process regression is considered as the potential non-parametric method throughout this paper. Similar definitions of the above mentioned cases can also be found in [5]. In literature, numeric methods have been used to solve the estimation problems in those cases. In [7], tracking based on Received-Signal-Strength (RSS) using a Gaussian Process-based measurement model is investigated. GP is also applied to model RSS measurements from inertial sensors in [15] and is further combined with distributed particle simultaneous localization and mapping to perform indoor tracking. Inference and learning is done in [6] for the case where both transition and measurement functions are modeled by Gaussian processes.

The CRBs have been derived extensively for case 0 in the literature. For instance, the parametric bound has been derived for discrete filtering in [16], and the posterior CRB for discrete filtering has been derived in [17] and [18]. However, for GP-based state-space models, to the best of our knowledge, there are rarely any relevant research. The technical difficulties for Gaussian process-based models may potentially occur in: computation of the Fisher/Bayesian information for the estimation, derive a statistical framework for Gaussian process state transition and measurement model, and evaluate the derived bounds with specified estimation examples. The contributions of this work can be summarized as follows: (I) state-space models based on Gaussian processes and corresponding state estimation algorithms are developed; (II) the theoretical lower bounds (i.e., parametric and posterior CRBs) for nonlinear state estimation problems are derived for Gaussian process-based state-space models; (III) furthermore, both parametric and posterior filtering CRBs are derived for a specific target tracking example; (IV) the derived CRBs and estimation algorithms are validated in simulations.

### C. Paper Organization and Notations

The remainder of this paper is organized as follows: A brief introduction to Gaussian process regression is given in Section II; Section III formulates three different state-space models based on Gaussian processes and provide corresponding estimation algorithms; Then, the parametric CRB is derived in Section IV, while the posterior CRBs for the three different cases are derived in Section V. CRBs for filtering are further derived for a specific example of target tracking in Section VI. The simulation results are given in Section VII. Finally, conclusions are drawn in Section VIII.

Throughout this paper, matrices are presented with uppercase letters, vectors with boldface lowercase letters and scalars with lowercase letters. The notation is defined and summarized in Table II.

## II. Brief Introduction of Gaussian Process

In this section, a brief introduction to Gaussian processes is provided. Gaussian process models constitute a class of important Bayesian non-parametric models for machine learning, which are tightly connected to several other salient models, such as support vector machines, single-layer Bayesian neural networks, and auto-regressive-moving-average [19]. The gist of GP models is to impose a Gaussian prior on the function/system $f(x)$ and then compute the posterior distribution over the function given the observed data. GP models have been used in a plethora of applications due to their outstanding performance in function approximation with a self-contained uncertainty bound. Gaussian process models are simple in terms of their mathematical formulation and analysis due to the Gaussian assumption.

To be more specific, a Gaussian process is a generalization of the Gaussian probability distribution. In a Gaussian process, every point in some continuous input space is associated with a normally distributed random variable. Moreover, every finite collection of those random variables has a multivariate normal distribution. The distribution of a Gaussian process is the joint distribution of all those (infinitely many) random variables.

Generally used as a machine learning method, the Gaussian process measures the similarity between different points to

TABLE II: Notations

| Notation | Definition |
|---|---|
| $[\cdot]^T$ | Vector/matrix transpose |
| $[\cdot]^{-1}$ | Inverse of a non-singular square matrix |
| $\mathrm{tr}(\cdot)$ | Trace of a square matrix |
| $I_n$ | Identity matrix of size $n$ |
| $\lVert \cdot \rVert$ | Euclidean norm of a vector |
| $\lvert \cdot \rvert$ | Cardinality of a set |
| $\mathbb{E}(\cdot)$ | Statistical expectation |
| $X(\cdot)^T$ | Short-hand notation for $XX^T$ |
| $\ln(\cdot)$ | Natural logarithm |
| $\log_{10}(\cdot)$ | Logarithm to base 10 |
| $\otimes$ | Kronecker product |
| $\nabla_{\boldsymbol{\theta}} = \partial/\partial\boldsymbol{\theta}$ | Gradient operator |
| $\Delta_{\boldsymbol{\theta}_2}^{\boldsymbol{\theta}_1} = \nabla_{\boldsymbol{\theta}_2}\nabla_{\boldsymbol{\theta}_1}^T$ | Hessian operator |
| $\mathcal{N}(\boldsymbol{x}\mid\boldsymbol{\mu}, \Sigma)$ | Gaussian distribution of $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and variance $\Sigma$ |
| $\boldsymbol{x}_{0:k}$ | Stacked vector from time 0 to time $k$, $[\boldsymbol{x}_0^T, \dots, \boldsymbol{x}_k^T]^T$ |

predict the observation value for new inputs. Let us consider a scalar real-valued Gaussian process regression model

$$y = f(\boldsymbol{x}) + e, \tag{1}$$

where $y$ is the scalar output, $\boldsymbol{x}$ is the input vector, $e$ is additive noise and $f(\boldsymbol{x})$ is a Gaussian process denoted

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{2}$$

The function $f(\boldsymbol{x})$ is characterized by its mean $m(\boldsymbol{x})$ and covariance/kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$. The mean and kernel function may contain a set of hyperparameters, which are denoted as $\boldsymbol{\theta}$. The noise term is usually considered to be Gaussian distributed with zero mean and variance $\sigma_e^2$. Examples of the covariance function include the stationary kernels, such as the Squared Exponential (SE), Matérn and exponential kernels [20], and also non-stationary kernels studied in [21], [22].

Usually, we are primarily interested in incorporating the knowledge from the training data set onto the function. Let us denote the training data set as $\mathcal{D} \triangleq \{\bar{X}, \bar{\boldsymbol{y}}\}$, where $\bar{X} = [\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_{N-1}]$ is the set of inputs and $\bar{\boldsymbol{y}} = [\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_{N-1}]^T$ is the set of outputs. Then we make predictions. It is natural to have the joint distribution of the training observations $\bar{\boldsymbol{y}}$, and the prediction of function $f^*$ at a new test input value $\boldsymbol{x}^*$. From previous statements about Gaussian processes, we know that the joint distribution of a collection of Gaussian distributed variables are still Gaussian distributed. Hence, we have

$$p(f|X, \theta) = \mathcal{N}(f|\boldsymbol{m}(\bar{X}), K(\bar{X}, \bar{X}, \boldsymbol{\theta})), \tag{3}$$

where $\boldsymbol{m}(\bar{X}) = [m(\bar{\boldsymbol{x}}_0), m(\bar{\boldsymbol{x}}_1), \ldots, m(\bar{\boldsymbol{x}}_{N-1})]^T$ and $K(\bar{X}, \bar{X}, \boldsymbol{\theta})$ denotes the covariance matrix for the training input data $\bar{X}$:

$$K(\bar{X}, \bar{X}, \boldsymbol{\theta}) = \begin{bmatrix} k(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_0) & \cdots & k(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_{N-1}) \\ \vdots & \ddots & \vdots \\ k(\bar{\boldsymbol{x}}_{N-1}, \bar{\boldsymbol{x}}_0) & \cdots & k(\bar{\boldsymbol{x}}_{N-1}, \bar{\boldsymbol{x}}_{N-1}) \end{bmatrix}. \tag{4}$$

For the noisy observations $\bar{\boldsymbol{y}}$ (the noise and the function $f$ are usually assumed to be uncorrelated), the probability distribution can be obtained by adding zero mean Gaussian noise of covariance $\sigma_e^2 I_N$, with $I_N$ denotes the identity matrix of size $N$, which is given by

$$p(\bar{\boldsymbol{y}}|\bar{X}, \boldsymbol{\theta}) = \mathcal{N}(\bar{\boldsymbol{y}}|\boldsymbol{m}(\bar{X}), K(\bar{X}, \bar{X}, \boldsymbol{\theta}) + \sigma_e^2 I_N). \tag{5}$$

Then, the joint distribution of the observations $\bar{\boldsymbol{y}}$ and a new function prediction $f^*$ at a new input $\boldsymbol{x}^*$ can be derived as

$$\begin{aligned} &p(f^*, \bar{\boldsymbol{y}}|\bar{X}, \boldsymbol{x}^*, \boldsymbol{\theta}) \\ &= \mathcal{N}\left(f^*, \bar{\boldsymbol{y}} \,\middle|\, \begin{bmatrix} \boldsymbol{m}(\bar{X}) \\ m(\boldsymbol{x}^*) \end{bmatrix}, \begin{bmatrix} K(\bar{X}, \bar{X}) + \sigma_e^2 I_N & \boldsymbol{k}(\bar{X}, \boldsymbol{x}^*) \\ \boldsymbol{k}(\boldsymbol{x}^*, \bar{X}) & k(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix}\right), \end{aligned} \tag{6}$$

where $\boldsymbol{k}(\bar{X}, \boldsymbol{x}^*) = [k(\bar{\boldsymbol{x}}_0, \boldsymbol{x}^*), \ldots, k(\bar{\boldsymbol{x}}_{N-1}, \boldsymbol{x}^*)]^T = \boldsymbol{k}^T(\boldsymbol{x}^*, \bar{X})$. Similarly, $k(\boldsymbol{x}^*, \boldsymbol{x}^*)$ is the covariance of $f(\boldsymbol{x}^*)$, and $K(\bar{X}, \bar{X})$ is a shorthand notation for $K(\bar{X}, \bar{X}, \boldsymbol{\theta})$. It should be noted that $K(\bar{X}, \bar{X})$, $\boldsymbol{k}(\bar{X}, \boldsymbol{x}^*)$ and $k(\boldsymbol{x}^*, \boldsymbol{x}^*)$ all contain the hyperparameter $\boldsymbol{\theta}$. Then, the prediction distribution, given all the training data and the test point $\boldsymbol{x}^*$, can be obtained according to the properties of multivariate Gaussian distribution

$$p(f^*|\boldsymbol{x}^*, \bar{X}, \bar{\boldsymbol{y}}, \boldsymbol{\theta}) = \mathcal{N}\left(f^*|\bar{\mu}^*, \mathrm{cov}(f^*)\right), \tag{7}$$

where

$$\bar{\mu}^* = m(\boldsymbol{x}^*) + \boldsymbol{k}(\boldsymbol{x}^*, \bar{X})[K(\bar{X}, \bar{X}) + \sigma_e^2 I_N]^{-1}(\bar{\boldsymbol{y}} - \boldsymbol{m}(\bar{X})), \tag{8a}$$

$$\mathrm{cov}(f^*) = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}(\boldsymbol{x}^*, \bar{X})[K(\bar{X}, \bar{X}) + \sigma_e^2 I_N]^{-1}\boldsymbol{k}(\bar{X}, \boldsymbol{x}^*). \tag{8b}$$

Detailed derivations can be found in [20, Appendix 2]. Then, the conditional probability of the noisy observation $y^*$ at the test point $\boldsymbol{x}^*$ can be derived as

$$\begin{aligned} p(y^*|\boldsymbol{x}^*, \bar{X}, \bar{\boldsymbol{y}}, \boldsymbol{\theta}) &= \int \mathcal{N}(y^*, f^*|\boldsymbol{x}^*, \bar{X}, \bar{\boldsymbol{y}}, \boldsymbol{\theta}) df^* \\ &= \int p(y^*|f^*) p(f^*|\boldsymbol{x}^*, \bar{X}, \bar{\boldsymbol{y}}, \boldsymbol{\theta}) df^* \\ &= \mathcal{N}(y^*|\bar{\mu}^*, \bar{k}^*), \end{aligned} \tag{9}$$

where

$$\begin{aligned} \bar{k}^* &= \mathrm{cov}(f^*) + \sigma_e^2 \\ &= k(\boldsymbol{x}^*, \boldsymbol{x}^*) + \sigma_e^2 \\ &\quad - \boldsymbol{k}(\boldsymbol{x}^*, \bar{X})[K(\bar{X}, \bar{X}) + \sigma_e^2 I_N]^{-1}\boldsymbol{k}(\bar{X}, \boldsymbol{x}^*). \end{aligned} \tag{10}$$

It is noted that for (9) to hold, both distributions $p(y^*|f^*)$ and $p(f^*|\boldsymbol{x}^*, \bar{X}, \bar{\boldsymbol{y}}, \boldsymbol{\theta})$ need to be Gaussian, which is fulfilled according to (7) and the model given in (1).

## III. Gaussian Process-based System Models

In this section, we will first introduce a general state-space model and briefly describe the estimation problem. Then, the system models for the last three cases defined in Table I are investigated.

We focus on an estimation problem which can be cast into a state-space formulation, which, in a general form is given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{w}_k, \tag{11a}$$

$$\boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{e}_k, \tag{11b}$$

where $\boldsymbol{x}_k$ denotes the $p \times 1$ state vector which needs to be estimated, $\boldsymbol{y}_k$ denotes the $n \times 1$ measurement vector, $\boldsymbol{w}_k$ is the process noise which is independent and Gaussian distributed with zero mean and covariance $Q$, $\boldsymbol{e}_k$ is the measurement noise which is also independent and zero mean Gaussian with covariance $R$. The state transition function is defined as $\boldsymbol{f}(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$, and the measurement function is defined as $\boldsymbol{g}(\cdot) : \mathbb{R}^p \to \mathbb{R}^n$. The functions $\boldsymbol{f}(\cdot)$, $\boldsymbol{g}(\cdot)$ are in arbitrary parametric/non-parametric forms. Both functions may depend on some unknown parameters, which are denoted by $\boldsymbol{\theta}$. It is further assumed that the parametric functions are smooth and continuously differentiable. For non-parametric functions, we assume that $\boldsymbol{f}(\boldsymbol{x})$ belongs to a reproducing kernel Hilbert space (RKHS) defined by the kernel function $K_f(\boldsymbol{x}, \boldsymbol{x}')$; Similarly, $\boldsymbol{g}(\boldsymbol{x})$ belongs to another RKHS defined by the kernel function $K_g(\boldsymbol{x}, \boldsymbol{x}')$. We need to assume further that both $K_f(\boldsymbol{x}, \boldsymbol{x}')$ and $K_g(\boldsymbol{x}, \boldsymbol{x}')$ are selected appropriately

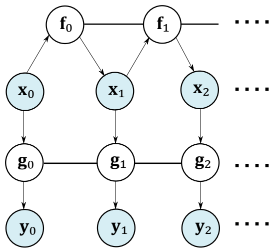

Fig. 1: Generic system model: $\boldsymbol{f}$ denotes the state transition function and $\boldsymbol{g}$ denotes the measurement model.

according to our prior knowledge about the data structure and they are smooth and differentiable. Universal kernels as introduced in [23] and [24] can be used if the prior knowledge is not available. If both functions are modeled by Gaussian processes, we obtain the graphical model as illustrated in Fig. 1, where there are dependencies over time (denoted by solid lines) between the transition functions and measurement functions [5]. It should be noted that this is not fully complying with the previously defined state-space model given in (11), which has the Markov property. Before deriving the theoretical lower bounds for the estimation problem formulated in (11), we first specify the following joint distribution

$$p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k}) = p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k}). \tag{12}$$

In what follows, $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})$ and $p(\boldsymbol{x}_{0:k})$ will be derived for three different cases.

To train the state transition and measurement models, a set of training data $\mathcal{D}$ has been collected during the off-line training phase. The training inputs $\bar{X}$ are the same as those defined in Section II. The training outputs consist of a stack of measurement vectors $\bar{\boldsymbol{y}} = [\bar{\boldsymbol{y}}_0^T, \bar{\boldsymbol{y}}_1^T, \ldots, \bar{\boldsymbol{y}}_{N-1}^T]^T$. After the off-line training phase, both the parameters in the parametric function and hyperparameters $\boldsymbol{\theta}$ in the GP model are estimated. In this work, the estimation error of the parameters are not considered in the bound calculation. Hence, when we derive the bounds, all the model parameters are considered as deterministically known. However, it would be interesting to include the estimation error of the model parameters in the bound derivation in our future research. Then, in the on-line estimation phase, we are aiming to estimate the current state vector $\boldsymbol{x}_k$, given the measurements up to time $k$, namely $\boldsymbol{y}_{0:k}$.

### *A. GP-based State Transition Model with GP-based Measurement Model*

In this case, we investigate the case where both state transition and measurement models are non-parametric Gaussian processes with additional white Gaussian noise. To be more specific, the functions $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ are defined as

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{m}_f(\boldsymbol{x}), K_f(\boldsymbol{x}, \boldsymbol{x}')), \tag{13}$$

$$\boldsymbol{g}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{m}_g(\boldsymbol{x}), K_g(\boldsymbol{x}, \boldsymbol{x}')), \tag{14}$$

where $\boldsymbol{m}_g(\boldsymbol{x}) = [m_{g,1}(\boldsymbol{x}), \ldots, m_{g,n}(\boldsymbol{x})]^T$ and $\boldsymbol{m}_f(\boldsymbol{x}) = [m_{f,1}(\boldsymbol{x}), \ldots, m_{f,p}(\boldsymbol{x})]^T$ are the mean functions. $K_g(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel/covariance function for the measurement model, which is a $n \times n$ square matrix and $K_f(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel function for the state transition model, which is a $p \times p$ square matrix. It is further assumed that the selected mean and kernel functions defined in (14) are smooth and continuously differentiable with respect to $\boldsymbol{x}$.

In general, to derive the theoretical lower bounds, we are aiming to compute the joint distribution of the states and measurements from time $0$ up to time $k$, given the training dataset $\mathcal{D}$, which can be formulated as

$$\begin{aligned} p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k}) &\triangleq p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k}|\mathcal{D}) \\ &= p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})p(\boldsymbol{x}_{0:k}|\mathcal{D}), \end{aligned} \tag{15}$$

where the subscript indicates a stack of vector from time 0 to $k$, $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})$ is the likelihood function and $p(\boldsymbol{x}_{0:k}|\mathcal{D})$ is the joint distribution of states given the training dataset.

Let us first derive the likelihood/conditional distribution of the measurements given the states from time 0 to $k$. As $\boldsymbol{g}(\boldsymbol{x})$ is a random function, the likelihood distribution is computed as

$$p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D}) = \int p(\boldsymbol{y}_{0:k}|\mathrm{g}_{0:k}, \mathcal{D})p(\mathrm{g}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})d\mathrm{g}_{0:k} \tag{16}$$

where $\mathrm{g}_{0:k} \triangleq [\boldsymbol{g}(\boldsymbol{x}_0)^T, \ldots, \boldsymbol{g}(\boldsymbol{x}_k)^T]^T$ denotes the stacked measurement functions from time 0 to $k$. For example, when $k = 0$, we have $\mathrm{g}_{0:0} = \boldsymbol{g}(\boldsymbol{x}_0)$. Given the functions $\mathrm{g}_{0:k}$, the measurement vector $\boldsymbol{y}_{0:k}$ is obtained by adding independent additive Gaussian noise. Hence, the distribution $p(\boldsymbol{y}_{0:k}|\mathrm{g}_{0:k}, \mathcal{D})$ is independent of the training dataset $\mathcal{D}$, yielding

$$\begin{aligned} p(\boldsymbol{y}_{0:k}|\mathrm{g}_{0:k}, \mathcal{D}) &= p(\boldsymbol{y}_{0:k}|\mathrm{g}_{0:k}) \\ &= \mathcal{N}(\boldsymbol{y}_{0:k}|\mathrm{g}_{0:k}, I_{k+1} \otimes R). \end{aligned} \tag{17}$$

It remains to derive the term $p(\mathrm{g}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})$, which can be computed according to the definition of Gaussian process as

$$p(\mathrm{g}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D}) = \mathcal{N}(\mathrm{g}_{0:k}|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k}), \bar{K}_g(\boldsymbol{x}_{0:k})), \tag{18}$$

where

$$\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k}) = K_g^T(\boldsymbol{x}_{0:k}, \bar{X})\tilde{K}_g^{-1}(\bar{\boldsymbol{y}} - \boldsymbol{m}_g(\bar{X})) + \boldsymbol{m}_g(\boldsymbol{x}_{0:k}) \tag{19a}$$

$$\bar{K}_g(\boldsymbol{x}_{0:k}) = K_g(\boldsymbol{x}_{0:k}, \boldsymbol{x}_{0:k}) - K_g^T(\boldsymbol{x}_{0:k}, \bar{X})\tilde{K}_g^{-1}K_g(\boldsymbol{x}_{0:k}, \bar{X}), \tag{19b}$$

with $\tilde{K}_g$ a short-hand notation for $\tilde{K}_g(\bar{X}, \bar{X}) \triangleq K_g(\bar{X}, \bar{X}) + I_N \otimes R$. The mean $\boldsymbol{m}_g(\bar{X})$ and covariance $K_g(\bar{X}, \bar{X})$ are constructed by the definition of a Gaussian process as

$$\boldsymbol{m}_g(\bar{X}) \triangleq \left[\boldsymbol{m}_g^T(\bar{\boldsymbol{x}}_0), \boldsymbol{m}_g^T(\bar{\boldsymbol{x}}_1), \ldots, \boldsymbol{m}_g^T(\bar{\boldsymbol{x}}_{N-1})\right]^T \tag{20a}$$

$$K_g(\bar{X}, \bar{X}) \triangleq \begin{bmatrix} K_g(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_0) & \cdots & K_g(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_{N-1}) \\ K_g(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_0) & \cdots & K_g(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_{N-1}) \\ \vdots & \ddots & \vdots \\ K_g(\bar{\boldsymbol{x}}_{N-1}, \bar{\boldsymbol{x}}_0) & \cdots & K_g(\bar{\boldsymbol{x}}_{N-1}, \bar{\boldsymbol{x}}_{N-1}) \end{bmatrix}. \tag{20b}$$

Similarly, $K_g(\boldsymbol{x}_{0:k}, \boldsymbol{x}_{0:k})$ and $K_g(\boldsymbol{x}_{0:k}, \bar{X})$ can be constructed as illustrated in (20b). It is worth noticing, that from (19b), $\bar{K}_g(\boldsymbol{x}_{0:k})$ has a non-diagonal structure due to the dependencies of the measurement functions $\boldsymbol{g}(\boldsymbol{x})$ over time. The diagonal elements of the matrix $\bar{K}_g(\boldsymbol{x}_{0:k})$ is given by

$$\bar{K}_g(\boldsymbol{x}_k) = K_g(\boldsymbol{x}_k, \boldsymbol{x}_k) - K_g^T(\boldsymbol{x}_k, \bar{X})\tilde{K}_g^{-1}K_g(\boldsymbol{x}_k, \bar{X}), \tag{21}$$

and the off-diagonal elements are given by

$$K_g(\boldsymbol{x}_k, \boldsymbol{x}_j) - K_g^T(\boldsymbol{x}_k, \bar{X})\tilde{K}_g^{-1}K_g(\boldsymbol{x}_j, \bar{X}), k \neq j. \tag{22}$$

The mean vector $\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k})$ is given by

$$\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k}) = [\bar{\boldsymbol{\mu}}_g^T(\boldsymbol{x}_0), \bar{\boldsymbol{\mu}}_g^T(\boldsymbol{x}_1), \ldots, \bar{\boldsymbol{\mu}}_g^T(\boldsymbol{x}_k)]^T, \tag{23}$$

where

$$\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k) = K_g^T(\boldsymbol{x}_k, \bar{X})\tilde{K}_g^{-1}(\bar{\boldsymbol{y}} - \boldsymbol{m}_g(\bar{X})) + \boldsymbol{m}_g(\boldsymbol{x}_k). \tag{24}$$

Then, the likelihood distribution can be computed by substituting (18) into (16), yielding

$$\begin{aligned} p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D}) &= \int p(\boldsymbol{y}_{0:k}|\mathbf{g}_{0:k})p(\mathbf{g}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})d\mathbf{g}_{0:k} \\ &= \int \mathcal{N}(\boldsymbol{y}_{0:k}|\mathbf{g}_{0:k}, I_{k+1}\otimes R)\mathcal{N}(\mathbf{g}_{0:k}|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k}), \bar{K}_g(\boldsymbol{x}_{0:k}))d\mathbf{g}_{0:k} \\ &= \mathcal{N}(\boldsymbol{y}_{0:k}|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_{0:k}), \bar{K}_g(\boldsymbol{x}_{0:k}) + I_{k+1} \otimes R). \end{aligned} \tag{25}$$

Furthermore, the joint distribution of the states can be partitioned as

$$\begin{aligned} p(\boldsymbol{x}_{0:k}|\mathcal{D}) &= \int p(\boldsymbol{x}_{0:k}, \mathbf{f}_{0:k-1}|\mathcal{D})d\mathbf{f}_{0:k-1} \\ &= \int p(\boldsymbol{x}_{0:k}|\mathbf{f}_{0:k-1}, \mathcal{D})p(\mathbf{f}_{0:k-1}|\mathcal{D})d\mathbf{f}_{0:k-1} \end{aligned} \tag{26}$$

where $\mathbf{f}_{0:k-1} \triangleq [\boldsymbol{f}(\boldsymbol{x}_0)^T, \ldots, \boldsymbol{f}(\boldsymbol{x}_{k-1})^T]^T$. For example, when $k = 1$, we have $\mathbf{f}_{0:0} = \boldsymbol{f}(\boldsymbol{x}_0)$. Given the trained functions $\boldsymbol{f}(\boldsymbol{x}_{0:k-1})$, the states at time $k$ can be obtained by adding independent Gaussian noise. Hence, the distribution $p(\boldsymbol{x}_{0:k}|\mathbf{f}_{0:k-1}, \mathcal{D})$ is independent of the training dataset $\mathcal{D}$ given the functions $\mathbf{f}_{0:k-1}$, yielding

$$\begin{aligned} p(\boldsymbol{x}_{0:k}|\mathbf{f}_{0:k-1}, \mathcal{D}) &= p(\boldsymbol{x}_{0:k}|\mathbf{f}_{0:k-1}) \\ &= p(\boldsymbol{x}_0)\prod_{t=1}^{k}\mathcal{N}(\boldsymbol{x}_t|\mathbf{f}_{t-1}, Q) \\ &= p(\boldsymbol{x}_0)\mathcal{N}(\boldsymbol{x}_{1:k}|\mathbf{f}_{0:k-1}, I_k \otimes Q). \end{aligned} \tag{27}$$

To train the state transition function, we take $\bar{X}_\mathrm{I} \triangleq [\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_{N-2}]$ and $\bar{\boldsymbol{x}}_\mathrm{O} \triangleq [\bar{\boldsymbol{x}}_1^T, \bar{\boldsymbol{x}}_2^T, \ldots, \bar{\boldsymbol{x}}_{N-1}^T]^T$ from the training dataset $\mathcal{D} = \{\bar{X}, \bar{\boldsymbol{y}}\}$ as the training inputs and outputs, respectively. From the introduction of Gaussian process given in Section II, we have

$$p(\mathbf{f}_{0:k-1}|\mathcal{D}) = \mathcal{N}(\mathbf{f}_{0:k-1}|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{0:k-1}), \bar{K}_f(\boldsymbol{x}_{0:k-1})), \tag{28}$$

where

$$\begin{aligned} \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{0:k-1}) &= K_f^T(\boldsymbol{x}_{0:k-1}, \bar{X}_\mathrm{I})\tilde{K}_f^{-1}(\bar{\boldsymbol{x}}_\mathrm{O} - \boldsymbol{m}_f(\bar{X}_\mathrm{I})) \\ &\quad + \boldsymbol{m}_f(\boldsymbol{x}_{0:k-1}) \end{aligned} \tag{29a}$$

$$\begin{aligned} \bar{K}_f(\boldsymbol{x}_{0:k-1}) &= K_f(\boldsymbol{x}_{0:k-1}, \boldsymbol{x}_{0:k-1}) \\ &\quad - K_f^T(\boldsymbol{x}_{0:k-1}, \bar{X}_\mathrm{I})\tilde{K}_f^{-1}K_f(\boldsymbol{x}_{0:k-1}, \bar{X}_\mathrm{I}), \end{aligned} \tag{29b}$$

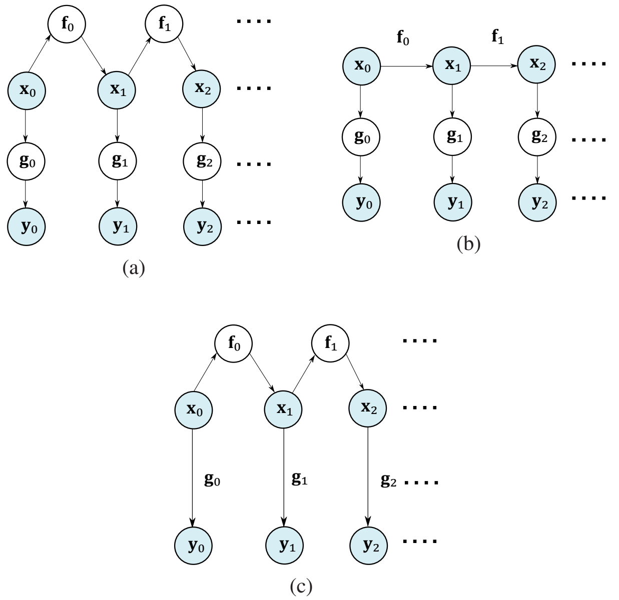

Fig. 2: State-space models: (a), GP-based state transition and measurement functions; (b), parametric state transition and GP-based measurement functions; (c), GP-based state transition and parametric measurement functions.

with $\tilde{K}_f$ stands for $\tilde{K}_f(\bar{X}_\mathrm{I}, \bar{X}_\mathrm{I}) \triangleq K_f(\bar{X}_\mathrm{I}, \bar{X}_\mathrm{I}) + I_{N-1} \otimes Q$. The mean $\boldsymbol{m}_f(\bar{X}_\mathrm{I})$ and covariance $K_f(\bar{X}_\mathrm{I}, X_\mathrm{I})$ are constructed by the definition of Gaussian process as

$$\boldsymbol{m}_f(\bar{X}_\mathrm{I}) \triangleq \left[\boldsymbol{m}_f^T(\bar{\boldsymbol{x}}_0), \boldsymbol{m}_f^T(\bar{\boldsymbol{x}}_1), \ldots, \boldsymbol{m}_f^T(\bar{\boldsymbol{x}}_{N-2})\right]^T \tag{30a}$$

$$K_f(\bar{X}_\mathrm{I}, \bar{X}_\mathrm{I}) \triangleq \begin{bmatrix} K_f(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_0) & \cdots & K_f(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{x}}_{N-2}) \\ K_f(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_0) & \cdots & K_f(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_{N-2}) \\ \vdots & \ddots & \vdots \\ K_f(\bar{\boldsymbol{x}}_{N-2}, \bar{\boldsymbol{x}}_0) & \cdots & K_f(\bar{\boldsymbol{x}}_{N-1}, \bar{\boldsymbol{x}}_{N-2}) \end{bmatrix}. \tag{30b}$$

In general, $\bar{K}_f(\boldsymbol{x}_{0:k-1})$ is not a diagonal matrix. The diagonal elements are given by

$$\bar{K}_f(\boldsymbol{x}_k) = K_f(\boldsymbol{x}_k, \boldsymbol{x}_k) - K_f^T(\boldsymbol{x}_k, \bar{X}_\mathrm{I})\tilde{K}_f^{-1}K_f(\boldsymbol{x}_k, \bar{X}_\mathrm{I}), \tag{31}$$

and the off-diagonal elements is given by

$$K_f(\boldsymbol{x}_k, \boldsymbol{x}_j) - K_f^T(\boldsymbol{x}_k, \bar{X}_\mathrm{I})\tilde{K}_f^{-1}K_f(\boldsymbol{x}_j, \bar{X}_\mathrm{I}), k \neq j. \tag{32}$$

The mean vector $\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{0:T})$ is given by

$$\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{0:k}) = [\bar{\boldsymbol{\mu}}_f^T(\boldsymbol{x}_0), \bar{\boldsymbol{\mu}}_f^T(\boldsymbol{x}_1), \ldots, \bar{\boldsymbol{\mu}}_f^T(\boldsymbol{x}_k)]^T, \tag{33}$$

where

$$\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_k) = K_f^T(\boldsymbol{x}_k, \bar{X}_\mathrm{I})\tilde{K}_f^{-1}(\bar{\boldsymbol{x}}_\mathrm{O} - \boldsymbol{m}_f(\bar{X}_\mathrm{I})) + \boldsymbol{m}_f(\boldsymbol{x}_k). \tag{34}$$

By substituting (28) and (27) into (26), we obtain

$$\begin{aligned} p(\boldsymbol{x}_{0:k}|\mathcal{D}) &= \int p(\boldsymbol{x}_{0:k}|\mathbf{f}_{0:k-1})p(\mathbf{f}_{0:k-1}|\mathcal{D})d\mathbf{f}_{0:k-1} \\ &= p(\boldsymbol{x}_0)\mathcal{N}(\boldsymbol{x}_{1:k}|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{0:k-1}), \bar{K}_f(\boldsymbol{x}_{0:k-1}) + I_k \otimes Q). \end{aligned} \tag{35}$$

Hence, the joint probability of all states and measurements up to time $k$ can be computed by substituting (35) and (25) into (15). However, by further examining the covariance matrix

$\bar{K}_g(\boldsymbol{x}_{0:k})$ and $\bar{K}_f(\boldsymbol{x}_{0:k-1})$, it is observed that $\boldsymbol{x}_k$ depends not only on $\boldsymbol{x}_{k-1}$, but also $\boldsymbol{x}_{0:k-2}$. Similarly, the measurement $\boldsymbol{y}_k$ depends on $\boldsymbol{x}_{0:k}$ and $\boldsymbol{y}_{0:k-1}$. The above mentioned correlations will further complicate the derivation of Cramér-Rao Bounds, which may lead to greatly increased computational complexity. Hence, the generic Gaussian process-based system model shown in Fig. 1 is further simplified by assuming independence across state transition functions $\mathbf{f}$ and measurement functions $\mathbf{g}$ as illustrated in Fig. 2a. One practical example would be a target tracking scenario, with the positions/velocity are considered as states, and the radio measurements are used as observations. Then, it is usually assumed that the current state is related to the state at previous time stamp, and the radio measurements are only related to current position. In other words, after we have trained the GP models off-line, at each time $k$, the distribution of measurement $\boldsymbol{y}_k$ can be computed given the input $\boldsymbol{x}_k$ and training data, yielding

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k, \mathcal{D}) = \mathcal{N}(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R). \tag{36}$$

Since the measurement at time $k$ only depends on current states $\boldsymbol{x}_k$ and training dataset, the likelihood function reduces to

$$\begin{aligned} p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D}) &= \prod_{t=0}^{k} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \mathcal{D}) && (37)\\ &= \prod_{t=0}^{k} \mathcal{N}(\boldsymbol{y}_t|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_t), \bar{K}_g(\boldsymbol{x}_t) + R). && (38) \end{aligned}$$

The joint distribution of all states up to time $k$ reduces to

$$\begin{aligned} &p(\boldsymbol{x}_{0:k}|\mathcal{D}) = p(\boldsymbol{x}_0) \prod_{t=1}^{k} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \mathcal{D}) && (39)\\ &= p(\boldsymbol{x}_0) \prod_{t=1}^{k} \int p(\boldsymbol{x}_t|\mathbf{f}_{t-1}) p(\mathbf{f}_{t-1}|\boldsymbol{x}_{t-1}, \mathcal{D}) d\mathbf{f}_{t-1}\\ &= p(\boldsymbol{x}_0) \prod_{t=1}^{k} \int \mathcal{N}(\boldsymbol{x}_t|\mathbf{f}_{t-1}, Q) \mathcal{N}(\mathbf{f}_{t-1}|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{t-1}), \bar{K}_f(\boldsymbol{x}_{t-1})) d\mathbf{f}_{t-1}\\ &= p(\boldsymbol{x}_0) \prod_{t=1}^{k} \mathcal{N}(\boldsymbol{x}_t|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{t-1}), \bar{K}_f(\boldsymbol{x}_{t-1}) + Q). && (40) \end{aligned}$$

### *B. Parametric State Transition Model with GP-based Measurement Model*

In this case, the state-space model appears as a parametric state transition with a non-parametric measurement model. Again, the model can be simplified as illustrated in Fig. 2b if further assuming independence across the measurement functions $\mathbf{g}$. Hence, the likelihood distribution $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})$ can be computed as given in (38).

Since the state transition function is deterministic and due to the independence assumption across the functions $\mathbf{f}$, the Markov property of the states is preserved. Hence, the joint distribution of state vectors up to time $k$ can be computed as

$$\begin{aligned} &p(\boldsymbol{x}_{0:k}) = p(\boldsymbol{x}_k|\boldsymbol{x}_{0:k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{x}_{0:k-2}) \cdots p(\boldsymbol{x}_1|\boldsymbol{x}_0) p(\boldsymbol{x}_0)\\ &= p(\boldsymbol{x}_0) \prod_{t=1}^{k} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p(\boldsymbol{x}_0) \prod_{t=1}^{k} \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{f}(\boldsymbol{x}_{t-1}), Q). && (41) \end{aligned}$$

### *C. GP-based State Transition Model with Parametric Measurement Model*

This subsection studies the case where the state transition is modeled by Gaussian process and the measurement model is given by a parametric and deterministic function $\boldsymbol{g}$. Using the same assumption as in (40), independence is assumed across functions $\boldsymbol{f}$. Since the measurement function $\boldsymbol{g}$ is deterministic and there is no correlation over time, the state-space model in this case can be graphically illustrated in Fig. 2c.

Without repeating similar derivations, the joint distribution of states up to time $k$ is given in (35). Considering a state-space model as illustrated in Fig. 2c, the likelihood distribution $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})$ is derived as

$$p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}) = \prod_{t=0}^{k} p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \prod_{t=0}^{k} \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{g}(\boldsymbol{x}_t), R). \tag{42}$$

### *D. Some Implications to State Estimation and Assumptions*

Considering the three different cases that have been discussed so far, some implications on the state estimation algorithm for each model will be summarized in this subsection. To be more specific, a particle filtering algorithm is adapted to the three different cases.

A particle filter numerically approximates the posterior density $p(\boldsymbol{x}_k|\boldsymbol{y}_{0:k})$ by a finite set of $N_s$ 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 [25]. Since its development in the early 1990s, various particle filter variants have appeared in the literature [26]. In this work, regarding the above mentioned GP-based state-space models, we focus on the bootstrap particle filter with importance density chosen to be the transitional prior $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$, due to its simplicity and satisfactory performance. The particle filtering algorithm for the estimation problem formulated previously is summarized in Algorithm 1.

For different state-space models as introduced in previous subsections, the state transitional prior and likelihood distribution are specified differently. Hence, we have

1) For Gaussian process-based state transition and Gaussian process-based measurement model,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &\triangleq p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathcal{D})\\ &= \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q)\\ p(\boldsymbol{y}_k|\boldsymbol{x}_k) &\triangleq p(\boldsymbol{y}_k|\boldsymbol{x}_k, \mathcal{D})\\ &= \mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right). \end{aligned}$$

where $\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)$ and $\bar{K}_g(\boldsymbol{x}_k)$ are given in (24) and (21), and $\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_k)$ and $\bar{K}_f(\boldsymbol{x}_k)$ are given in (34) and (31).

2) For parametric state transition and Gaussian process-based measurement model,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &= \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{f}(\boldsymbol{x}_{k-1}), Q)\\ p(\boldsymbol{y}_k|\boldsymbol{x}_k) &\triangleq p(\boldsymbol{y}_k|\boldsymbol{x}_k, \mathcal{D})\\ &= \mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right). \end{aligned}$$

where $\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)$ and $\bar{K}_g(\boldsymbol{x}_k)$ are given in (24) and (21).

**Algorithm 1** Particle Filter

```
1) Initialization: Draw $N_s$ samples $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$, and set $w_0^{(i)} = 1/N_s$, for all $i = 1, ..., N_s$.
2) For $k = 1, ..., T$, do:
    a) For $i = 1, ..., N_s$, draw samples from the state transition distribution,
           $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)}).$        (43)
    b) Update weights according to
           $w_k^{(i)} \propto w_{k-1}^{(i)} p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}),$
       for $i = 1, ..., N_s$, such that $\sum_{i=1}^{N_s} w_k^{(i)} = 1$.
    c) An approximation to the posterior filtering expectation is given by
           $\hat{\boldsymbol{x}}_{\text{PF},k}(\boldsymbol{y}_{0:k}) \triangleq \sum_{i=1}^{N_s} w_k^{(i)} \boldsymbol{x}_k^{(i)} \approx \mathbb{E}\{\boldsymbol{x}_k|\boldsymbol{y}_{0:k}\}.$
    d) Resampling: If $N_{\text{eff}} = 1/\sum_{i=1}^{N}(w_k^{(i)})^2 < N_{\text{th}}$, then perform multinomial resampling [27].
```

3) For Gaussian process-based state transition and parametric measurement model,

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &\triangleq p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathcal{D}) \\ &= \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q) \\ p(\boldsymbol{y}_k|\boldsymbol{x}_k) &= \mathcal{N}(\boldsymbol{y}_k|\boldsymbol{g}(\boldsymbol{x}_k), R), \end{aligned}$$

where $\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_k)$ and $\bar{K}_f(\boldsymbol{x}_k)$ are given in (34) and (31).

Before we start to derive the CRBs, there are some assumptions that we need to state. In this work, we consider a relatively simpler GP system model in the sense that a batch of training data is available a priori. In one example, the data set $\mathcal{D}$ can be a set of calibrated trajectories with known inputs (states) and output (e.g., wireless measurements). With this training data set, we can get the posterior distribution (i.e., the equations $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ and $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ given above), given the GP prior $\boldsymbol{f}(\boldsymbol{x}) \sim GP(\boldsymbol{m}(\boldsymbol{x}), K_f(\boldsymbol{x}, \boldsymbol{x}'))$ and the data set $\mathcal{D}$. Take $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ as an example, the posterior can be seen as $\boldsymbol{x}_k = \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k$, with the noise $\boldsymbol{v}_k$ covariance matrix equal to $\bar{K}_f(\boldsymbol{x}_{k-1}) + Q$. When the training data set is large, the GP modeling error $\bar{K}_f(\boldsymbol{x}_{k-1})$ will be negligible (i.e., the model parameters are considered as deterministically known). When the training data set is too small, $\bar{K}_f(x_{k-1})$ will be large and even dominant in comparison with $Q$, and in this case the bound will be affected.

However, in [2], [28], they assume no prior training data, and the GP model (namely the kernel hyperparameters) will be estimated jointly with the latent state in the GP system model. The estimation problem is very difficult. The log-marginal likelihood itself is mathematically intractable so that the Evidence Lower BOund (ELBO) is derived instead using a variational approximation. The estimation problem involves a large scale optimization which is not guaranteed to converge to a stationary point. Deriving a bound for this more advanced model would be very interesting in our future work.

## IV. Parametric CRB

Parametric CRB provides a lower bound of the mean squared error (MSE) matrix, conditioned on a specific state sequence $\boldsymbol{x}_{0:k}$. Due to the conditioning, there is no need to generate state trajectories from the process model, and one can use *any* state trajectory of interest. The parametric CRB for filtering $P_{0:k}(\boldsymbol{x}_{0:k})$ bounds on the conditional MSE matrix for any unbiased estimator $\hat{\boldsymbol{x}}_{0:k}(\boldsymbol{y}_{0:k})$ as follows

$$\mathbb{E}\{(\hat{\boldsymbol{x}}_{0:k}(\boldsymbol{y}_{0:k}) - \boldsymbol{x}_{0:k})(\cdot)^T \mid \boldsymbol{x}_{0:k}\} \succeq P_{0:k}(\boldsymbol{x}_{0:k}), \tag{47}$$

where $P_{0:k}(\boldsymbol{x}_{0:k})$ is the inverse of the auxiliary Fisher information matrix (FIM) $J_{0:k}(\boldsymbol{x}_{0:k})$ [16], $(X)(\cdot)^T$ denotes $XX^T$, and the inequality means that the difference $\mathbb{E}\{(\hat{\boldsymbol{x}}_{0:k}(\boldsymbol{y}_{0:k}) - \boldsymbol{x}_{0:k})(\cdot)^T \mid \boldsymbol{x}_{0:k}\} - P_{0:k}(\boldsymbol{x}_{0:k})$ is positive semi-definite. The parametric CRB only holds for unbiased estimators.

The auxiliary Fisher information matrix $J_{0:k}(\boldsymbol{x}_{0:k})$ of the state sequence $\boldsymbol{x}_{0:k}$ is defined as

$$J_{0:k}(\boldsymbol{x}_{0:k}) = \mathbb{E}_{\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}}\{-\Delta_{\boldsymbol{x}_{0:k}}^{\boldsymbol{x}_{0:k}} \ln p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})|\boldsymbol{x}_{0:k}\} \tag{48}$$

The auxiliary sequence $\boldsymbol{z}_{1:k}$ is introduced to incorporate the deterministic state information into the bound computations. We rewrite this as a set of equality constraints, given by: $\boldsymbol{x}_{i+1} - \boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{w}_i = 0$, for $i = 0, \ldots, k-1$. Then, it can be further interpreted as a set of measurements $\boldsymbol{x}_{i+1} - \boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{w}_i = \boldsymbol{z}_{i+1}$, with $\boldsymbol{z}_{i+1} = 0, \forall i$. To stay in the probabilistic framework, those equality constraints are added as zero-mean Gaussian noise with covariance $\Pi$, where the covariance will be set to zero to recover the equality constraints [29]. In this section, we derive the parametric filtering bound for case 1, where the state transition is formulated by deterministic parametric functions. Since the parametric bound is computed for a specific realization of the state trajectory, when the state transition $\boldsymbol{f}$ is modeled by non-parametric function (i.e., there is no deterministic function form), the parametric bound can not be computed.

There are some regularity conditions which are required for the parametric CRB to exist, as given in [30], [31] and [32]:

1) The distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$ has to be differentiable with respect to $\boldsymbol{z}_{1:k}$.
2) The support of the distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$ on $\boldsymbol{y}_{0:k}$ does not depend on $\boldsymbol{z}_{1:k}$.

We further verify that both conditions are fulfilled with the model provided in III-B.

1) The distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$, with $\boldsymbol{z}_{1:k}$ being the auxiliary sequence, can be written as $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k}) = p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$. From the definition of the auxiliary sequence, it is easy to verify that $p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$ is a multivariate Gaussian distribution. Hence, $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$ is differentiable with respect to $\boldsymbol{z}_{1:k}$.
2) The support $\mathbb{S}$ of the distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$ is the support of the Gaussian distribution function $p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_{0:k})$. Hence, $\mathbb{S} = \mathbb{R}^n$, which does not depend on $\boldsymbol{z}_{1:k}$.

In this case, at time $k$, the auxiliary Fisher information submatrix $J_k(\boldsymbol{x}_{0:k})$ [16] is computed as the inverse of the $p \times p$

lower-right partition of $J_{0:k}^{-1}(\boldsymbol{x}_{0:k})$, which can be computed recursively as

$$J_k(\boldsymbol{x}_{0:k}) = D_k^{22} - \left(D_k^{12}\right)^T \left[D_k^{11} + J_{k-1}(\boldsymbol{x}_{0:k-1})\right]^{-1} D_k^{12}. \tag{49}$$

where $J_{k-1}(\boldsymbol{x}_{0:k-1})$ is the auxiliary Fisher information submatrix at time $k-1$ and

$$D_k^{11} = \mathbb{E}_{\boldsymbol{z}_k}\left\{-\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_{k-1}} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{x}_{k-1})|\boldsymbol{x}_{0:k}\right\} \tag{50a}$$

$$D_k^{12} = \mathbb{E}_{\boldsymbol{z}_k}\left\{-\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{x}_{k-1})|\boldsymbol{x}_{0:k}\right\} \tag{50b}$$

$$\begin{aligned} D_k^{22} = {}& \mathbb{E}_{\boldsymbol{z}_k}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{x}_{k-1})|\boldsymbol{x}_{0:k}\right\} \\ &+ \mathbb{E}_{\boldsymbol{y}_k}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln p(\boldsymbol{y}_k|\boldsymbol{x}_k)|\boldsymbol{x}_{0:k}\right\}. \end{aligned} \tag{50c}$$

The detailed derivations are provided in a supplementary document. Considering the system model given in Section III-B, the $D$-terms can be further simplified to

$$\begin{aligned} D_k^{11} &= \mathbb{E}_{\boldsymbol{z}_k}\{-\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_{k-1}} \ln \mathcal{N}(\boldsymbol{z}_k|\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1}) - \boldsymbol{w}_{k-1}, \Pi)|\boldsymbol{x}_{0:k}\} \\ &= F_{k-1}^T(\boldsymbol{x}_{k-1})\Pi^{-1}F_{k-1}(\boldsymbol{x}_{k-1}) \end{aligned} \tag{51a}$$

$$\begin{aligned} D_k^{12} &= \mathbb{E}_{\boldsymbol{z}_k}\left\{-\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{z}_k|\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1}) - \boldsymbol{w}_{k-1}, \Pi)|\boldsymbol{x}_{0:k}\right\} \\ &= F_{k-1}^T(\boldsymbol{x}_{k-1})\Pi^{-1} \end{aligned} \tag{51b}$$

$$\begin{aligned} D_k^{22} &= \mathbb{E}_{\boldsymbol{z}_k}\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{z}_k|\boldsymbol{x}_k - f(\boldsymbol{x}_{k-1}) - \boldsymbol{w}_{k-1}, \Pi)|\boldsymbol{x}_{0:k}\} \\ &+ \mathbb{E}_{\boldsymbol{y}_k}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right)|\boldsymbol{x}_{0:k}\right\} \\ &= \Pi^{-1} + G_k \end{aligned} \tag{51c}$$

where

$$F_{k-1}(\boldsymbol{x}_{k-1}) = \left[\nabla_{\boldsymbol{x}_{k-1}} f^T(\boldsymbol{x}_{k-1})\right]^T \tag{52}$$

and the term $G_k$ is defined as

$$G_k \triangleq \mathbb{E}_{\boldsymbol{y}_k}\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R)|\boldsymbol{x}_k\}. \tag{53}$$

According to [31, page 47], the element at row $i$ column $j$ in $G_k$, can be computed as

$$\begin{aligned} [G_k]_{i,j} = {}& \left[\frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial x_{k,i}}\right]^T \hat{K}_g^{-1}(\boldsymbol{x}_k) \left[\frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial x_{k,j}}\right] \\ &+ \frac{1}{2}\text{tr}\left[\hat{K}_g^{-1}(\boldsymbol{x}_k)\frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,i}}\hat{K}_g^{-1}(\boldsymbol{x}_k)\frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,j}}\right], \end{aligned} \tag{54}$$

with $\hat{K}_g(\boldsymbol{x}) \triangleq \bar{K}_g(\boldsymbol{x}) + R$.

Now, we can further simplify the recursive expression for $J_k(\boldsymbol{x}_{0:k})$ by substituting the $D$-terms

$$\begin{aligned} J_k(\boldsymbol{x}_{0:k}) = {}& G_k + \Pi^{-1} \\ & - \Pi^{-1}F_{k-1}\left[F_{k-1}^T\Pi^{-1}F_{k-1} + J_{k-1}\right]^{-1} F_{k-1}^T\Pi^{-1}, \end{aligned} \tag{55}$$

where $F_{k-1}$ stands for $F_{k-1}(\boldsymbol{x}_{k-1})$ and $J_{k-1}$ is a shorthand notation for $J_{k-1}(\boldsymbol{x}_{0:k-1})$. Then, using matrix inversion lemma the following can be obtained

$$J_k(\boldsymbol{x}_{0:k}) = \left(\Pi + F_{k-1}J_{k-1}^{-1}F_{k-1}^T\right)^{-1} + G_k. \tag{56}$$

To recover the equality constraints in the state sequence, $\Pi$ needs to be set to zero, yielding

$$J_k(\boldsymbol{x}_{0:k}) = \left(F_{k-1}J_{k-1}^{-1}F_{k-1}^T\right)^{-1} + G_k. \tag{57}$$

**Algorithm 2** Parametric Filtering CRB

```
1) Assuming the prior distribution is Gaussian $p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0; \boldsymbol{\mu}_{\boldsymbol{x}_0}, \Sigma_{\boldsymbol{x}_0})$, initialize the FIM with $J_0(\boldsymbol{x}_0) = \Sigma_{\boldsymbol{x}_0}^{-1}$. The parametric filtering CRB is computed as $P_0(\boldsymbol{x}_0) = J_0^{-1}(\boldsymbol{x}_0)$.
2) For $k = 0, \dots, T$, do:
    • Compute the filtering FIM according to [16]
          $J_k = \left(F_{k-1}J_{k-1}^{-1}F_{k-1}^T\right)^{-1} + G_k,$
      where $F_{k-1}(\boldsymbol{x}_{k-1}) = \left[\nabla_{\boldsymbol{x}_{k-1}} f^T(\boldsymbol{x}_{k-1})\right]^T$ and $G_k$ is given in (53) and (54).
    • Compute the parametric filtering CRB as
      $P_k(\boldsymbol{x}_{0:k}) = J_k^{-1}(\boldsymbol{x}_{0:k})$.
```

With above derivations, the parametric filtering CRB for GP-based measurement model can be computed recursively as summarized in Algorithm 2. Although case 0 as described in the introduction section is a simple case, which has been explored extensively (e.g., see [16]), we would like to mention that the parametric CRB for case 0 is similar to what has been derived above, except that $G_k$ would be replaced by $[\nabla_{\boldsymbol{x}_k}\boldsymbol{g}^T(\boldsymbol{x}_k)]R^{-1}[\nabla_{\boldsymbol{x}_k}\boldsymbol{g}^T(\boldsymbol{x}_k)]^T$.

## V. Posterior CRB

Assessing the fundamental performance limitations in Bayesian filtering is usually carried out using the posterior (or Bayesian) CRB. In Bayesian filtering, the posterior CRB $P_{\text{B}}$ puts a lower bound on the MSE matrix of any estimator $\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{0:k})$, i.e.

$$\mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}}\left\{(\hat{\boldsymbol{x}}_{0:k}(\boldsymbol{y}_{0:k}) - \boldsymbol{x}_{0:k})(\cdot)^T\right\} \succeq P_{\text{B},0:k}, \tag{58}$$

where $P_{\text{B},0:k}$ is the inverse of the Bayesian information matrix (BIM) $J_{\text{B},0:k}$ and $(X)(\cdot)^T$ denotes $XX^T$. Unlike the parametric CRB, the estimator in the above inequality is not required to be unbiased. Compared to parametric CRB, which gives the theoretical lower bound for a specific state trajectory, the posterior CRB takes the average bounds over all possible state trajectories and measurements. According to [17], the Bayesian information matrix $J_{\text{B},0:k}$ of the state sequence $\boldsymbol{x}_{0:k}$ is defined as

$$J_{\text{B},0:k} = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}}\left\{-\Delta_{\boldsymbol{x}_{0:k}}^{\boldsymbol{x}_{0:k}} \ln p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k})\right\}. \tag{59}$$

There are also some regularity conditions to meet for the posterior CRB to exist according to [30] and [17], where the distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k})$ should be differentiable with respect to $\boldsymbol{x}_{0:k}$. This can be verified for all three cases given in Section III.

1) For case 1, the state transition is parametric and the measurement model is given by Gaussian process. The distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k}) = p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})p(\boldsymbol{x}_{0:k})$, where $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})$ and $p(\boldsymbol{x}_{0:k})$ are given in (25) and (35), respectively. From the definition of the parametric function $\boldsymbol{f}(\boldsymbol{x})$ at the beginning of Section III, it is differentiable with respect to $\boldsymbol{x}$. For the Gaussian process

**Algorithm 3** Posterior Filtering CRB

```
1) Initialize the BIM with $J_{\mathrm{B},0}$. The posterior filtering CRB
   is computed as $P_{\mathrm{B},0} = J_{\mathrm{B},0}^{-1}$.
2) For $k = 0, \ldots, T$, do:
   • Compute the filtering BIM from
       $J_{\mathrm{B},k} = L_k^{22} - L_k^{21} \left[ L_k^{11} + J_{\mathrm{B},k-1} \right]^{-1} L_k^{12}$,   (60)
     where $L_k^{11}$, $L_k^{12}$, $L_k^{21}$ and $L_k^{22}$ are given by
       $L_k^{11} = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_{k-1}} \ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \right\}$   (61a)
       $L_k^{12} = \left(L_k^{21}\right)^T = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \right\}$   (61b)
       $L_k^{22} = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \right\}$
               $+ \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln p(\boldsymbol{y}_k|\boldsymbol{x}_k) \right\}$   (61c)
     where $L_k^{11}$, $L_k^{12}$ and $L_k^{22}$ for different models are
     given in subsections V-A to V-C.
   • Compute the posterior filtering CRB as $P_{\mathrm{B},k} = J_{\mathrm{B},k}^{-1}$.
```

$\boldsymbol{g}(\boldsymbol{x})$, both the mean and kernel functions are differentiable with respect to $\boldsymbol{x}$ as defined in Section III-A. From (25), $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k}, \mathcal{D})$ is a Gaussian function with both the mean and covariance differentiable with respect to $\boldsymbol{x}_{0:k}$. Hence, it is differentiable with respect to $\boldsymbol{x}_{0:k}$. The distribution function $p(\boldsymbol{x}_{0:k})$ in (35) is a product of Gaussian distributions and the initial distribution of $\boldsymbol{x}_0$, which is usually assumed to be Gaussian as well. Hence, $p(\boldsymbol{x}_{0:k})$ are also differentiable with respect to $\boldsymbol{x}_{0:k}$.

2) For case 2, the distribution function $p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k}) = p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})p(\boldsymbol{x}_{0:k}|\mathcal{D})$. From (42), $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})$ is a product of Gaussian distributions, and the parametric function $\boldsymbol{g}(\boldsymbol{x})$ is defined as differentiable with respect to $\boldsymbol{x}$ at the beginning of Section III. Hence, $p(\boldsymbol{y}_{0:k}|\boldsymbol{x}_{0:k})$ is differentiable with respect to $\boldsymbol{x}_{0:k}$. For $p(\boldsymbol{x}_{0:k}|\mathcal{D})$ in (40), it is a product of Gaussian distributions and both the mean and kernel functions of the Gaussian process $\boldsymbol{f}(\boldsymbol{x})$ are differentiable with respect to $\boldsymbol{x}$ as defined in Section III-A. Hence, $p(\boldsymbol{x}_{0:k}|\mathcal{D})$ is differentiable with respect to $\boldsymbol{x}_{0:k}$, and finally $p(\boldsymbol{y}_{0:k}, \boldsymbol{x}_{0:k})$ is differentiable with respect to $\boldsymbol{x}_{0:k}$.
3) For case 3, the condition has been verified from case 1 and 2.

From the definition, expectation with respect to all state trajectories and measurements needs to be computed. A closed form of the posterior CRB is available for linear, additive Gaussian state-space models. In most of the nonlinear case, Monte Carlo evaluations are needed to simulate all possible state trajectories and measurements.

Given the state-space models formulated in Section III, the BIM can be computed recursively as given in Algorithm 3. Detailed derivations are given in the supplementary document. In what follows, the $L$-terms for the previously defined three cases are derived.

### A. Posterior Filtering CRB: Case 1

For state-space model as illustrated in Fig. 2b, we can further derive the $L$-terms, yielding

$$\begin{aligned} L_k^{11} &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_{k-1}} \ln \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{f}(\boldsymbol{x}_{k-1}), Q) \right\} \\ &= \mathbb{E}_{\boldsymbol{x}_{k-1}} \left\{ F_{k-1}^T(\boldsymbol{x}_{k-1}) Q^{-1} F_{k-1}(\boldsymbol{x}_{k-1}) \right\} \qquad (62a) \\ L_k^{12} &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{f}(\boldsymbol{x}_{k-1}), Q) \right\} \\ &= \mathbb{E}_{\boldsymbol{x}_{k-1}} \left\{ F_{k-1}^T(\boldsymbol{x}_{k-1}) \right\} Q^{-1} \qquad (62b) \\ L_k^{22} &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{x}_k|\boldsymbol{f}(\boldsymbol{x}_{k-1}), Q) \right\} \\ &\quad + \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right) \right\} \\ &= Q^{-1} + M_k, \qquad (62c) \end{aligned}$$

where $F_{k-1}(\boldsymbol{x}_{k-1})$ is defined in (52), and

$$M_k \triangleq \mathbb{E}_{\boldsymbol{y}_k,\boldsymbol{x}_k} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right) \right\}, \tag{63}$$

and the element at row $i$ column $j$ in $M_k$, based on [31, page 47], is given as

$$\begin{aligned} [M_k]_{i,j} &= \mathbb{E}_{\boldsymbol{x}_k} \left\{ \left[ \frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial x_{k,i}} \right]^T \hat{K}_g^{-1}(\boldsymbol{x}_k) \left[ \frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial x_{k,j}} \right] \right\} \\ &+ \mathbb{E}_{\boldsymbol{x}_k} \left\{ \frac{1}{2} \mathrm{tr} \left[ \hat{K}_g^{-1}(\boldsymbol{x}_k) \frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,i}} \hat{K}_g^{-1}(\boldsymbol{x}_k) \frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,j}} \right] \right\}, \end{aligned} \tag{64}$$

with $\hat{K}_g(\boldsymbol{x}) \triangleq \bar{K}_g(\boldsymbol{x}) + R$. Note that the expectations are not always tractable, and when they are intractable, Monte Carlo integrations are usually applied to compute the expectations, see for instance, [30] and [18]. Theoretically, by taking Monte Carlo integrations, this will not be an absolute bound. However, if the number of Monte Carlo runs are sufficiently large, the Monte Carlo integrations will converge to the exact bounds. For all practical situations, the values obtained from Monte Carlo integrations can be close enough to the true ones.

### B. Posterior Filtering CRB: Case 2

The BIM for the system in Fig. 2c can be iteratively computed according to (60), and the $L$-terms are derived in this subsection. To compute the $L$-terms, we first need to specify the distribution $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. From (40) and (39), we have

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &\triangleq p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \mathcal{D}) \\ &= \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q). \end{aligned} \tag{65}$$

Hence, the $L$-terms defined in (61) are derived as follow

$$L_k^{11} = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_{k-1}} \ln \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q) \right\} \tag{66a}$$

$$\begin{aligned} L_k^{12} &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_{k-1}}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q) \right\} \\ &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ \nabla_{\boldsymbol{x}_{k-1}} (\boldsymbol{x}_k - \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}))^T (\bar{K}_f(\boldsymbol{x}_{k-1}) + Q)^{-1} \right\} \end{aligned} \tag{66b}$$

$$\begin{aligned} L_k^{22} &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1}) + Q) \right\} \\ &+ \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ -\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k} \ln \mathcal{N}\left(\boldsymbol{y}_k|\boldsymbol{g}(\boldsymbol{x}_k), R\right) \right\} \\ &= \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ (\bar{K}_f(\boldsymbol{x}_{k-1}) + Q)^{-1} \right\} \\ &+ \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}} \left\{ [\nabla_{\boldsymbol{x}_k} \boldsymbol{g}^T(\boldsymbol{x}_k)] R^{-1} [\nabla_{\boldsymbol{x}_k} \boldsymbol{g}^T(\boldsymbol{x}_k)]^T \right\}. \end{aligned} \tag{66c}$$

According to [31, page 47], the element at row $i$ column $j$ in $L_k^{11}$, is given by

$$[L_k^{11}]_{i,j} = \mathbb{E}_{\boldsymbol{x}_{k-1}}\left\{\left[\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}\right]^T \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}) \left[\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,j}}\right]\right\}$$
$$+\mathbb{E}_{\boldsymbol{x}_{k-1}}\left\{\frac{1}{2}\mathrm{tr}\left[\hat{K}_f^{-1}(\boldsymbol{x}_{k-1})\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}\hat{K}_f^{-1}(\boldsymbol{x}_{k-1})\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,j}}\right]\right\}, \tag{67}$$

where $\hat{K}_f(\boldsymbol{x}) \triangleq \bar{K}_f(\boldsymbol{x}) + Q$ and $\bar{K}_f(\boldsymbol{x})$ is given in (31). The derivative in $L_k^{12}$ can be expressed as

$$\nabla_{\boldsymbol{x}_{k-1}}(\boldsymbol{x}_k - \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}))^T(\bar{K}_f(\boldsymbol{x}_{k-1}) + Q)^{-1}$$
$$\triangleq \nabla_{\boldsymbol{x}_{k-1}}\boldsymbol{\phi}(\boldsymbol{x}_{k-1}) \tag{68}$$

where the $i$-th element in $\nabla_{\boldsymbol{x}_{k-1}}\boldsymbol{\phi}(\boldsymbol{x}_{k-1})$, $i = 1, \ldots, p$, is given by

$$\frac{\partial \boldsymbol{\phi}(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}} = -\hat{K}_f^{-1}(\boldsymbol{x}_{k-1})\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$$
$$+\hat{K}_f^{-1}(\boldsymbol{x}_{k-1})\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}\hat{K}_f^{-1}(\boldsymbol{x}_{k-1})\left[\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}) - \boldsymbol{x}_k\right] \tag{69}$$

### C. Posterior Filtering CRB: Case 3

In this subsection, the $L$-terms for the system model in Fig. 2a are derived. To compute the $L$-terms, we need to specify the likelihood distribution $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ and the state transition probability $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. From (38) and (37), (40) and (39), we can easily obtain the distributions. Hence, the $L$-terms defined in (61) are derived as follow: $L_k^{11}$ is given in (66a) and (67), $L_k^{12}$ is given in (66b) and (69), $L_k^{22}$ is derived as

$$L_k^{22} = \mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\ln\mathcal{N}(\boldsymbol{x}_k|\bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \bar{K}_f(\boldsymbol{x}_{k-1})+Q)\right\}$$
$$+\mathbb{E}_{\boldsymbol{y}_{0:k},\boldsymbol{x}_{0:k}}\left\{-\Delta_{\boldsymbol{x}_k}^{\boldsymbol{x}_k}\ln\mathcal{N}\left(\boldsymbol{y}_k|\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k), \bar{K}_g(\boldsymbol{x}_k) + R\right)\right\}$$
$$= \mathbb{E}_{\boldsymbol{x}_{k-1}}\left\{(\bar{K}_f(\boldsymbol{x}_{k-1}) + Q)^{-1}\right\} + M_k, \tag{70a}$$

where $M_k$ is given in (64).

## VI. Application Example

In this section, we aim to compute the parametric and posterior CRBs for a practical example according to the derivations given in Section IV and V.

In this application example, a typical indoor target tracking scenario is considered, where the tracking is perfomed by estimating the positions iteratively based on the dynamic state transition (also known as motion model) and measurement model. The measurements for this tracking problem are selected as the received-signal-strength, which is usually used as an indication of relative distance to the reference network nodes with known positions. It is assumed that the RSS measurements from different reference network nodes are independent of each other. It is also assumed elements in the state vector (positions and possibly velocities) are independent. Furthermore, independences across dynamic motion and RSS measurement functions over time are assumed. In what follows, the parametric and posterior CRBs are derived for this specific example.

### A. Parametric CRB for Target Tracking Example

In this case, the state transition function is deterministic and the measurement model is given by a Gaussian process. Considering the relative static indoor environment and low walking speed, we use a nearly constant velocity model with white noise as given in [33, Chapter 6] for the state transition,

$$\boldsymbol{x}_k = F\boldsymbol{x}_{k-1} + \boldsymbol{w}_{k-1}, \tag{71}$$

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

$$F = I_2 \otimes \begin{bmatrix} 1 & T_\mathrm{s} \\ 0 & 1 \end{bmatrix}, \tag{72}$$

where $T_\mathrm{s}$ is the sampling time. The vector $\boldsymbol{w}_k$ is assumed to be zero-mean Gaussian distributed with covariance matrix

$$Q = I_2 \otimes \begin{bmatrix} T_\mathrm{s}^3/3 & T_\mathrm{s}^2/2 \\ T_\mathrm{s}^2/2 & T_\mathrm{s} \end{bmatrix}\sigma_w^2, \tag{73}$$

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

The RSS measurement is formulated as a Gaussian process. Assuming that $\boldsymbol{y}_k$ consists of RSS measurements from different reference network nodes, and further assume that the RSS measurements from different nodes are independent. For measurement from the $i$-th node, the mean function is selected as

$$m_{g,i}(\boldsymbol{x}) = A_i + B_i \log_{10}||\mathbf{p}_{\boldsymbol{x}} - \bar{\mathbf{p}}_i||, i = 1, \ldots, n, \tag{74}$$

since the magnitude of RSS measurements usually decays logarithmically with distance (between the transmitter and receiver) [34], where $A_i$ is the transmit power at 1 meter away from the $i$-th reference node, $B_i$ is the path-loss exponent, and $\bar{\mathbf{p}}_i$ denotes the known position of $i$-th reference node.

The kernel function indicates the spatial correlation, which is usually assumed to follow Gudmundson's model [35],

$$k_{g,i}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{g,i}^2 \exp\left(-\frac{||\mathbf{p}_{\boldsymbol{x}} - \mathbf{p}_{\boldsymbol{x}'}||}{l_{g,i}^2}\right), i = 1, \ldots, n. \tag{75}$$

Hence, in order to compute the parametric CRB as summarized in Algorithm 2, we need to compute $F_{k-1}$ and $G_k$, where

$$F_{k-1} = \left[\nabla_{\boldsymbol{x}_{k-1}} f^T(\boldsymbol{x}_{k-1})\right]^T = F. \tag{76}$$

In order to compute $G_k$, it is required to further specify the terms $\frac{\partial \bar{\mu}_g(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_{k,i}}$ and $\frac{\partial \bar{K}_g(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_{k,i}}$, for $i = 1, \ldots, 4$. With $\bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)$ given in (24), we have

$$\frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_{k,i}} = \left(\frac{\partial K_g(\boldsymbol{x}_k, X)}{\partial x_{k,i}}\right)^T \tilde{K}_g^{-1}(\boldsymbol{y} - \boldsymbol{m}_g(X))$$
$$+ \frac{\partial \boldsymbol{m}_g(\boldsymbol{x}_k)}{x_{k,i}}, \tag{77}$$

where the element at row $j \times q$ column $j$ in $\frac{\partial K_g(\boldsymbol{x}_k, X)}{\partial x_{k,i}}$ is given by

$$\begin{aligned}
&\frac{\partial k_{g,j}(\boldsymbol{x}_k, \boldsymbol{x}_q)}{\partial x_{k,i}} \\
&= -\frac{\sigma_{g,j}^2 (x_{k,i} - x_{q,i})}{l_{g,j}^2 ||\mathbf{p}_{\boldsymbol{x}_k} - \mathbf{p}_{\boldsymbol{x}_q}||} \exp\left( -\frac{||\mathbf{p}_{\boldsymbol{x}_k} - \mathbf{p}_{\boldsymbol{x}_q}||}{l_{g,j}^2} \right), i = 1, 3.
\end{aligned} \tag{78}$$

$$\frac{\partial k_{g,j}(\boldsymbol{x}_k, \boldsymbol{x}_q)}{\partial x_{k,i}} = 0, i = 2, 4.$$

Furthermore, the $j$-th element in $\frac{\partial \boldsymbol{m}_g(\boldsymbol{x}_k)}{x_{k,i}}$ is given by

$$\frac{\partial m_{g,j}(\boldsymbol{x}_k)}{\partial x_{k,i}} = \frac{B_j}{\ln 10} \frac{(x_{k,i} - \bar{\mathrm{p}}_{j,i})}{||\mathbf{p}_{\boldsymbol{x}_k} - \bar{\mathbf{p}}_j||^2}, i = 1, 3 \tag{79a}$$

$$\frac{\partial m_{g,j}(\boldsymbol{x}_k)}{\partial x_{k,i}} = 0, i = 2, 4. \tag{79b}$$

The second term $\frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_{k,i}}$ is computed according to

$$\begin{aligned}
\frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,i}} &= -\frac{\partial \left( K_g^T(\boldsymbol{x}_k, X) \tilde{K}_g^{-1} K_g(\boldsymbol{x}_k, X) \right)}{\partial x_{k,i}} \\
&= -2 K_g^T(\boldsymbol{x}_k, X) \tilde{K}_g^{-1} \left[ \frac{\partial K_g(\boldsymbol{x}_k, X)}{\partial x_{k,i}} \right],
\end{aligned} \tag{80}$$

where $\frac{\partial K_g(\boldsymbol{x}_k, X)}{\partial x_{k,i}}$ is computed according to (78).

### *B. Posterior CRB for Target Tracking Example: Case 1*

Considering the system model in Fig. 2b, the computation of the posterior bound can be simplified by specifying the $L$-terms as

$$L_k^{11} = F^T Q^{-1} F, \tag{81a}$$

$$L_k^{12} = F^T Q^{-1}. \tag{81b}$$

To compute $L_k^{22}$, we need to compute the expectations. However, in many cases, it is not feasible to compute the expectations analytically. Hence, Monte Carlo simulations are usually applied to approximate the results. Monte Carlo integration is used to obtain the element at row $i$ column $j$ in $M_k$, which is given by

$$\begin{aligned}
[M_k]_{i,j} \approx \sum_{m=1}^{M} \frac{1}{M} \left[ \frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k^{(m)})}{\partial x_{k,i}} \right]^T \hat{K}_g^{-1}(\boldsymbol{x}_k^{(m)}) \left[ \frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k^{(m)})}{\partial x_{k,j}} \right] \\
+ \frac{1}{2M} \mathrm{tr} \left[ \hat{K}_g^{-1}(\boldsymbol{x}_k^{(m)}) \frac{\partial \hat{K}_g(\boldsymbol{x}_k^{(m)})}{\partial x_{k,i}} \hat{K}_g^{-1}(\boldsymbol{x}_k^{(m)}) \frac{\partial \hat{K}_g(\boldsymbol{x}_k^{(m)})}{\partial x_{k,j}} \right],
\end{aligned} \tag{82}$$

where $M$ is the number of Monte Carlo runs and $\boldsymbol{x}_k^{(m)}$ is the state trajectory sampled from the distribution $p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}) = \mathcal{N}(\boldsymbol{x}_k | F \boldsymbol{x}_{k-1}, Q)$. The terms $\frac{\partial \bar{\boldsymbol{\mu}}_g(\boldsymbol{x}_k)}{\partial x_{k,i}}$ and $\frac{\partial \hat{K}_g(\boldsymbol{x}_k)}{\partial x_{k,i}}$ are given in (77) and (80), respectively. Finally, we have

$$L_k^{22} = Q^{-1} + M_k, \tag{83}$$

with $M_k$ given in (82).

### *C. Posterior CRB for Target Tracking Example: Case 2*

In this subsection, the system model defined in Section III-C is adopted in the tracking example, where the state dynamic is formulated by Gaussian process and the RSS measurement model is deterministic. To be more specific, the state vector $\boldsymbol{x}_k$ only consists of the 2-D location $\boldsymbol{x}_k = \mathbf{p}_{\boldsymbol{x}_k} = [p_{\mathrm{x}}, p_{\mathrm{y}}]$, and it is common to assume that the positions in two dimensions are independent. Hence, each element in the state vector follows a Gaussian process, characterized by a zero mean function and an SE kernel function

$$k_{f,j}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{f,j}^2 \exp\left( -\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2 l_{f,j}^2} \right), j = 1, \ldots, p. \tag{84}$$

The measurement model is formulated deterministically using the log-distance model

$$g_i(\boldsymbol{x}_k) = A_i + B_i \log_{10} ||\boldsymbol{x}_k - \bar{\mathbf{p}}_i||, i = 1, \ldots, n, . \tag{85}$$

To compute the posterior CRB as derived in Section V-B, the $L$-terms are required to be specified, which leads to the derivatives $\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$ and $\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$, for $i = 1, \ldots, p$. With $\boldsymbol{m}_f(\boldsymbol{x}_{k-1}) = \mathbf{0}$ and the kernel function given in (84), we have

$$\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}} = \left[ \frac{\partial K_f(\boldsymbol{x}_{k-1}, X_{\mathrm{I}})}{\partial x_{k-1,i}} \right]^T \tilde{K}_f^{-1} \boldsymbol{x}_{\mathrm{O}}. \tag{86}$$

Then, we have the element at row $j \times q$ column $j$ in $\frac{\partial K_f(\boldsymbol{x}_{k-1}, X_{\mathrm{I}})}{\partial x_{k-1,i}}$

$$\begin{aligned}
&\frac{\partial k_{f,j}(\boldsymbol{x}_{k-1}, \boldsymbol{x}_q)}{\partial x_{k-1,i}} = \\
&- \frac{\sigma_{f,j}^2 (x_{k,i} - x_{q,i})}{l_{f,j}^2} \exp\left( -\frac{||\boldsymbol{x}_{k-1} - \boldsymbol{x}_q||^2}{2 l_{f,j}^2} \right).
\end{aligned} \tag{87}$$

Next, we compute the term $\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$, which is derived as

$$\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}} = -2 \left[ K_f^T(\boldsymbol{x}_{k-1}, X_I) \tilde{K}_f^{-1} \frac{\partial K_f(\boldsymbol{x}_{k-1}, X_I)}{\partial x_{k-1,i}} \right], \tag{88}$$

with $\frac{\partial K_f(\boldsymbol{x}_{k-1}, X_I)}{\partial x_{k-1,i}}$ computed from (87). Hence, $L_k^{11}$ can be approximated using Monte Carlo simulations

$$\begin{aligned}
L_k^{11} \approx \sum_{m=1}^{M} \frac{1}{M} \left\{ \left[ \frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}^{(m)})}{\partial x_{k-1,i}} \right]^T \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}^{(m)}) \left[ \frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}^{(m)})}{\partial x_{k-1,j}} \right] \right\} \\
+ \left\{ \frac{1}{2M} \mathrm{tr} \left[ \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}^{(m)}) \frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1}^{(m)})}{\partial x_{k-1,i}} \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}^{(m)}) \frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1}^{(m)})}{\partial x_{k-1,j}} \right] \right\},
\end{aligned} \tag{89}$$

where the state trajectory is sampled from the distribution $\mathcal{N}(\boldsymbol{x}_k | \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}), \hat{K}_f(\boldsymbol{x}_{k-1}))$. To compute $L_k^{12}$, we need to compute $\frac{\partial \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$ according to (86) and $\frac{\partial \hat{K}_f(\boldsymbol{x}_{k-1})}{\partial x_{k-1,i}}$ can be computed according to (88). Finally, $L_k^{12}$ can be evaluated by Monte Carlo simulations

$$L_k^{12} \approx \sum_{m=1}^{M} \frac{1}{M} \nabla_{\boldsymbol{x}_{k-1}} (\boldsymbol{x}_k^{(m)} - \bar{\boldsymbol{\mu}}_f(\boldsymbol{x}_{k-1}^{(m)}))^T \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}^{(m)}). \tag{90}$$

The last term to compute the posterior CRB is $L_k^{22}$, which can be approximately obtained as

$$\begin{aligned} L_k^{22} \approx & \sum_{m=1}^{M} \frac{1}{M} \hat{K}_f^{-1}(\boldsymbol{x}_{k-1}^{(m)}) \\ & + \frac{1}{M} \left( \nabla_{\boldsymbol{x}_k} \boldsymbol{g}(\boldsymbol{x}_k^{(m)}) \right)^T R^{-1} \nabla_{\boldsymbol{x}_k} \boldsymbol{g}(\boldsymbol{x}_k^{(m)}), \end{aligned} \tag{91}$$

where $\nabla_{\boldsymbol{x}_k} \boldsymbol{g}(\boldsymbol{x}_k)$ is a $p \times n$ matrix and the element at row $i$ column $j$ is given by

$$\frac{\partial g_j(\boldsymbol{x}_k)}{\partial x_{k,i}} = \frac{B_j}{\ln 10} \frac{(x_{k,i} - \bar{\mathrm{p}}_{j,i})}{||\boldsymbol{x}_k - \bar{\mathbf{p}}_j||^2}. \tag{92}$$

### D. Posterior CRB for Target Tracking Example: Case 3

In this subsection, the posterior CRB for system model with both state transition and measurement formulated by Gaussian processes is derived. Correspondingly, we have the state vector $\boldsymbol{x}_k$ consists of 2-D positions, while the measurement vector $\boldsymbol{y}_k$ contains RSS measurements from different reference network nodes. Hence, the mean and kernel functions for the state transition and measurement models are given by

$$m_{f,j}(\boldsymbol{x}) = 0, j = 1, \ldots, p \tag{93a}$$

$$k_{f,j}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{f,j}^2 \exp\left( -\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2 l_{f,j}^2} \right), j = 1, \ldots, p \tag{93b}$$

$$m_{g,i}(\boldsymbol{x}) = A_i + B_i \log_{10} ||\boldsymbol{x} - \bar{\mathbf{p}}_i||, i = 1, \ldots, n \tag{93c}$$

$$k_{g,i}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_{g,i}^2 \exp\left( -\frac{||\boldsymbol{x} - \boldsymbol{x}'||}{l_{g,i}^2} \right), i = 1, \ldots, n. \tag{93d}$$

From previous derivations, to compute the posterior CRB in this case, the $L$-terms are specified as follows

1) Compute $L_k^{11}$ according to (89);
2) Compute $L_k^{12}$ according to (90);
3) Compute $L_k^{22}$ according to (70), with $M_k$ given in (82) and the first expectation is computed using Monte Carlo integration.

## VII. Results

In this section, we first evaluate the derived CRBs with synthetic dataset. Then, the derived bounds are computed and compared to the estimation errors in a real tracking scenario.

### A. Synthetic Dataset Evaluations

Before we evaluate the derived CRBs in the tracking scenario, it is worth while to have some intuitive understanding and verifications about the CRBs derived in previous sections. Hence, we use a set of synthetic data and compute the posterior CRBs for case 3, where both state transition and measurement models are formulated by Gaussian processes, since this is the most interesting and complex case. We begin with a scalar case, and the results can be easily extended to higher dimensions based on the derivations provided in Section IV, V and VI. Hence, we have the following system model:

$$\begin{aligned} x_k &= f(x_{k-1}) + w_k \\ y_k &= g(x_k) + e_k, \end{aligned} \tag{94}$$

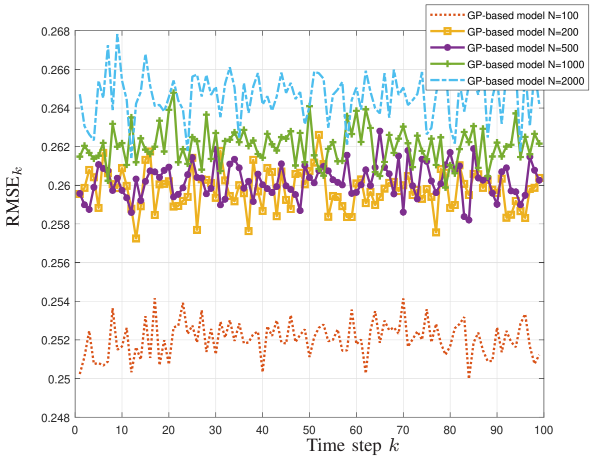

Fig. 3: The posterior filtering CRBs, Case 3 with synthetic data set.

where both $f(x_{k-1})$ and $g(x_k)$ are Gaussian processes with $m_f = m_g = 0$ and the SE kernels, and $w_k$ and $e_k$ are independent zero mean Gaussian noise with variance 1. To train the GP models, we need to generate a set of training data. Different numbers of training data are used, for instance, $N = 100, 200, 500, 1000, 2000$. After the training phase, we obtain a set of parameters, and those parameters are used to compute the posterior CRBs according to derivations in previous sections. The computed CRBs for case 3 with different numbers of training data are plotted in Fig. 3. From the results we see that as the number of training dataset increases, the derived bounds get tighter. However, for $N \geq 200$, the increase becomes insignificant. As the computational complexity to train the GP model and compute the bound increases greatly as $N$ becomes larger, we propose to choose the number of training data as $200 \leq N \leq 1000$.

### B. Real Data Collection and Simulation Setup

In this subsection, the parametric and posterior CRBs are evaluated in a tracking scenario given an office layout. In total $n = 12$ Blue-tooth low energy (BLE) beacons are used as reference network nodes with known positions and placed in the area such that a generally good geometry is obtained. The floor plan with beacon locations are shown in two-dimensional (2-D) space in Fig. 4 with a local coordinate system. The BLE beacons serve as transmitters and broadcast information regularly. A set of training data is collected along predefined tracks during normal work hours. During the data collection phase, the mobile device monitors the BLE advertisement channel and measures RSS. A total of 12 144 RSS measurements with corresponding positions[1] were collected, from which the propagation parameters $\{A_j, B_j, \sigma_j^2\}_{j=1}^n$, noise variance and the hyperparameters for Gaussian processes were estimated using a training data set of size $N = 600$. In this practical example, the states are defined as positions, velocities. The

[1] The positions are assumed to be precisely known and they were essentially obtained from an app-based positioning algorithm developed by Senion with an accuracy of 1 to 3 meters. [36]

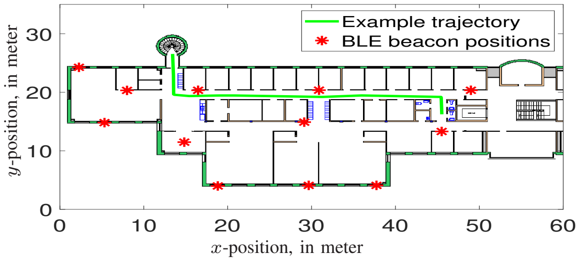

Fig. 4: Sensor deployment (with BLE beacons marked by red star) and one example of selected trajectory.

measurements are also predefined with proper dimensions, for instance, the dimension of the RSS measurements is determined by the number of beacons that are used for positioning. Hence, in the sequential estimation phase, the dimensions of the states and measurements are not changing.

After the propagation parameters and the hyperparameters are trained, the state transition and measurement model are used to simulate the true state trajectories and corresponding measurements. The parametric and posterior CRBs are then computed accordingly. The filtering algorithms as provided in Section III-D are applied to obtain the position estimates.

### C. Performance Metrics

For parametric CRB, in case the estimator $\hat{\boldsymbol{x}}_k(\boldsymbol{y}_{1:k})$ is conditionally unbiased, the position root conditional mean squared error (RCMSE) has the following relation

$$\text{RCMSE}_k \geq \sqrt{[P_k]_{1,1} + [P_k]_{3,3}} \triangleq \gamma_k, \tag{95}$$

where $[P_k]_{i,j}$ denotes the element at row $i$ column $j$ of the parametric CRB matrix $P_k(\boldsymbol{x}^*_{0:k})$. The RCMSE can be approximately computed as $\text{RCMSE}_k \approx \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\delta_k^{(m)}\right)^2}$, where $\delta_k^{(m)}$ is the position estimation error computed at each time step $k$ in the $m$-th Monte Carlo run, which can be computed as $\delta_k = ||\hat{\mathbf{p}}_{\boldsymbol{x}_k} - \mathbf{p}^*_{\boldsymbol{x}_k}||$, where $\hat{\mathbf{p}}_{\boldsymbol{x}_k}$ denotes the estimated position and $\mathbf{p}^*_{\boldsymbol{x}_k}$ is the ground truth.

The parametric CRB requires that the estimator is conditionally unbiased, a condition that is rarely met in practice. 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:

$$\boldsymbol{b}_k = [b_{\text{x},k}, b_{\text{y},k}]^T \approx \frac{1}{M}\sum_{m=1}^{M}\left[\hat{\mathbf{p}}_{\boldsymbol{x}_k^{(m)}} - \mathbf{p}^*_{\boldsymbol{x}_k}\right]. \tag{96}$$

The posterior CRBs relate to the position root mean squared error (RMSE) of the estimator as

$$\text{RMSE}_k \geq \sqrt{\text{trace}(P_{\text{B},k})} \triangleq \beta_k, \tag{97}$$

and the RMSE is given by $\text{RMSE}_k \approx \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\delta_k^{(m)}\right)^2}$, where $\delta_k^{(m)} = ||\hat{\mathbf{p}}_{\boldsymbol{x}_k^{(m)}} - \mathbf{p}^*_{\boldsymbol{x}_k^{(m)}}||$, and $\mathbf{p}^*_{\boldsymbol{x}_k^{(m)}}$ is the ground truth of the $m$-th state trajectory at time $k$, $\hat{\mathbf{p}}_{\boldsymbol{x}_k^{(m)}}$ is the position estimate at time $k$ for the $m$-th state trajectory.

TABLE III: Evaluation parameters.

| Parameter | Value | Description |
|---|---|---|
| $\sigma_w$ | 1 | Noise power spectral density |
| $T_s$ | 0.1 | Sampling interval |
| $N_s$ | 1000 | Number of particles |
| $n$ | 12 | Number of deployed reference nodes |
| $M$ | 10000 | Number of Monte Carlo runs |
| $N_{\text{th}}$ | $2/3\,N_s$ | Threshold for resampling |

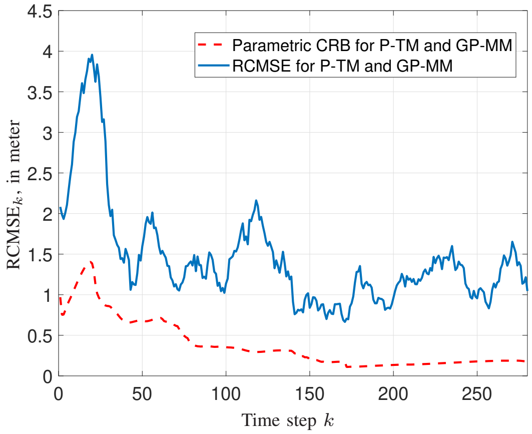

Fig. 5: RCMSE and parametric filtering CRB.

### D. Simulation Results

In this subsection, the positioning performance evaluated by the metrics defined previously are compared to parametric and posterior CRBs. The simulation parameters are listed in Table III. Different number of particles have been tested, where $N_s = 1000$ yields good trade off between the computation time and estimation performance. Similarly, the noise power spectral density in the nearly constant velocity model is selected such that good results are obtained.

The positioning accuracy of the state trajectory illustrated in Fig. 4 is compared to the parametric filtering CRBs in Fig. 5. It is observed that the RCMSE curve is above the parametric filtering bound. It is also noticed that the gap between the estimation performance in terms of $\text{RCMSE}_k$ and the bound is around 0.5 to 1 meter. The parametric bound holds for unbiased estimators as discussed in Section IV. However, this is merely met in practice. Hence, in the simulations, we compute the bias of the position estimator. By observing the bias shown in Fig. 6, it is noted that the bias in both $x$ and $y$-axes is high for $k$ between 20 and 30 and the bias in $y$-axes is high for $k$ between 100 and 150. Despite the bias, there is about 1 meter gap between the bound and the $\text{RCMSE}_k$. Hence, finding an estimator that reduces the estimation error worth further investigation.

The posterior CRBs are further compared to the RMSEs in three cases. For the first case with the Parametric state Transition Model (P-TM) and GP-based Measurement Model (GP-MM), the RMSE and the posterior CRB are depicted in Fig. 7. Since, the state transition model is given by the nearly constant velocity model, it generates state trajectories without

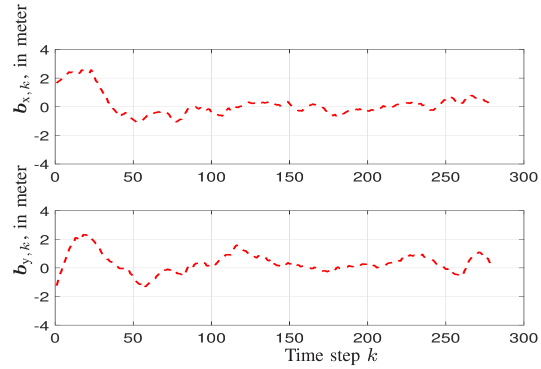

Fig. 6: Bias in position estimates.

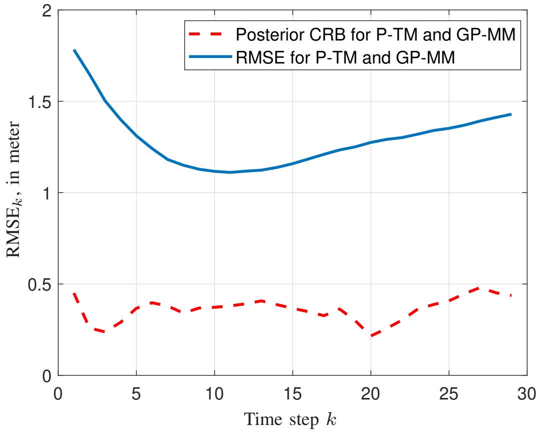

Fig. 7: RMSE and posterior filtering CRB: Case 1.

confining to the indoor environment constraints, such as the inner and outer walls. As time increases, more state trajectories will end up outside the office area without coverage of the deployed network nodes. Considering this factor, much shorter trajectories are simulated such that most of them remain within the office area. The RMSE follows the bound closely with a gap around 1 meter. For such a system with nearly constant velocity state transition and GP-based measurement model, the position estimation accuracy is above 1 meter, which is relatively far away from the theoretical lower limit at around 0.5 meter. Since more trajectories go out of the coverage area of the network nodes, the positioning error grows as the time increases.

For the second case, we have Gaussian Process-based State Transition model (GP-TM) and Parametric Measurement Model (P-MM). The RMSE and posterior CRB are illustrated in Fig. 8. The GP model for generating the state trajectories are trained off-line using trajectories that are collected within the office area. Hence, compared with the nearly constant velocity model where trajectories can potentially end up outside the building, the simulated state trajectories using GP are constrained within the office area. It is also clear that the estimation accuracy is above the posterior CRB. By applying GP to the state transition model, the position RMSE (around 0.75 meter) goes down and gets closer to the theoretical bound.

For the third case, where Gaussian process-based state transition model and Gaussian process-based measurement model are formulated, the RMSE and posterior CRB are illustrated in Fig. 9. By applying GP, more accurate models for state transitions are obtained. In addition, the relationships between RSS measurements and positions are more accurately represented by a GP. Hence, the position estimation accuracy has been improved (to around 0.6 meter), and the theoretical lower bound goes to around 0.4 meter.

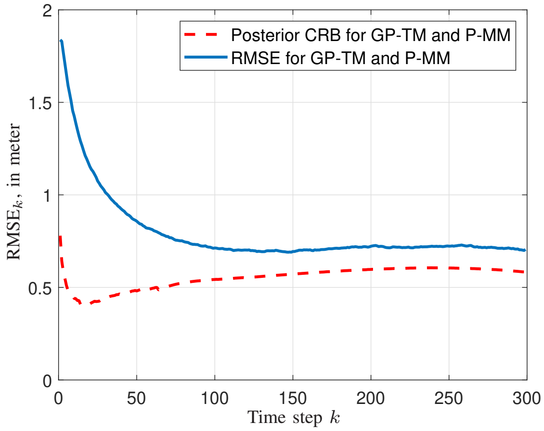

Fig. 8: RMSE and posterior filtering CRB: Case 2.

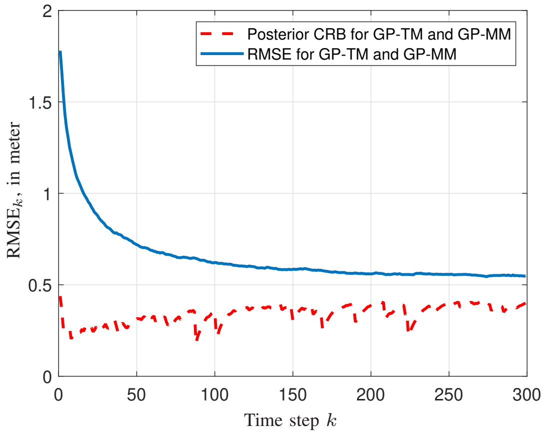

Fig. 9: RMSE and posterior filtering CRB: Case 3.

It is noted that the results show that the derived CRBs are strict lower bounds for this specific target tracking problem where particle filter is applied to perform the state estimation. The derived CRBs are considered tight in general (with a small gap, i.e., 0.2 to 1 meter, between the RMSE and the bounds). A tighter bound (e.g., only 0.2 meter gap between the estimation error and the lower bound) is obtained when both the state transition and measurement models are based on Gaussian process. The focus of this work is on derivation of the CRBs and finding an estimator that can achieve the bounds is out of the scope. However, it would be of great interest to investigate other estimation algorithms that could potentially get closer or even achieve the theoretical lower limits.

## VIII. Conclusions

The parametric and posterior filtering Cramér-Rao bounds have been derived for state-space models which are formulated or partly formulated by Gaussian processes. Three cases are investigated depending on which part of the model is given by Gaussian process. Further, a practical example has been illustrated, where Gaussian processes are applied to model the state transition and/or measurement model. Simulations are carried out considering a practical tracking problem in an office environment. By comparing the positioning accuracy with the CRBs, we conclude that the derived parametric and posterior CRBs can be used to evaluate non-linear estimation performance based on Gaussian processes.

In the derivations, we assume an off-line training phase and all model parameters are known in the on-line phase. However, it would be interesting to further investigate the effects of parameter estimation errors and derive the corresponding lower bounds. To look forward even further, it would be beneficial to study the case where the model parameters are also considered as unknown, and are estimated together with the state variables. This leads to the possible future research on deriving the hybrid CRBs for GP-based dynamic estimation problems.

In addition, although for GP-based state transition and measurement model, the estimation performance gets closer to the bound compared with other cases, it is interesting to further investigate the estimation algorithms where the performance can get closer or even achieve the theoretical lower bounds.

Finally, deriving a bound for the more advanced GP-based model as given in [2] and [28] would be very interesting as our future work.

## Acknowledgment

This work is funded by ELLIT project from 2017 to 2019, which is a strategic research environment funded by the Swedish government in 2010. This work is also funded by European Union FP7 Marie Curie training programme on Tracking in Complex Sensor Systems (TRAX) with grant number 607400 from 2014 to 2017. Furthermore, we acknowledge the support from Senion, who provides BLE beacons as well as associated positioned RSS measurement data.

Feng Yin is mainly funded by Shenzhen Science and Technology Innovation Council under Grant JCYJ20170307155957688, NSFC fund 61701426, Guangdong Province Pearl River Talent Team under Grant 2017ZT07X152, and partly by the Shenzhen Fundamental Research Fund under Grant (Key Lab) ZDSYS201707251409055. Tianshi Chen is funded by Shenzhen Science and Technology Innovation Council under Grant JCYJ20170411102101881.

## References

[1] Y. Zhao, C. Fritsche, and F. Gunnarsson, "Parametric lower bound for nonlinear filtering based on Gaussian process regression model," in *20th Int. Conf. on Information Fusion (FUSION)*, Xi'an, China, July 2017, pp. 1–7.

[2] R. Frigola, F. Lindsten, T. Schön, and C. E. Rasmussen, "Identification of Gaussian process state-space models with particle stochastic approximation EM," in *IFAC Proceedings Volumes (IFAC)*, vol. 19, 08, 2014, pp. 4097–4102.

[3] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," in *Int. Conf. on Intelligent Robots and Systems (IROS)*, Nice, France, Sep. 2008, pp. 3471–3476.

[4] P. Richter and M. Toledano-Ayala, "Revisiting Gaussian process regression modeling for localization in wireless sensor networks," *Sensors (Basel, Switzerland)*, vol. 15, no. 9, pp. 22 587–22 615, Sep. 2015.

[5] R. Frigola, "Bayesian time series learning with Gaussian processes," Ph.D. dissertation, University of Cambridge, 2015.

[6] R. Turner, M. Deisenroth, and C. Rasmussen, "State-space inference and learning with Gaussian processes," in *The 13th Int. Conf. on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 9. Sardinia, Italy: PMLR, May 2010, pp. 868–875.

[7] B. Ferris, D. Hähnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Robotics: Science and Systems (RSS)*, Philadelphia, USA, August 2006, pp. 1–8.

[8] Y. Zhao, F. Yin, F. Gunnarsson, M. Amirijoo, and G. Hendeby, "Gaussian processes for propagation modeling and proximity based indoor positioning," in *IEEE 83rd Vehicular Technology Conference: VTC2016-Spring*, Nanjing, China, May 2016, pp. 1–5.

[9] J. Wagberg, D. Zachariah, T. Schön, and P. Stoica, "Prediction Performance After Learning in Gaussian Process Regression," in *The 20th Int. Conf. on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54. Fort Lauderdale, FL, USA: PMLR, Apr 2017, pp. 1264–1272.

[10] F. Yin, Y. Zhao, F. Guannarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using Gaussian processes," *Transactions on Signal Processing*, vol. 65, no. 8, pp. 2164–2177, 2017.

[11] F. Yin, Y. Zhao, and F. Gunnarsson, "Fundamental bounds on position estimation using proximity reports," in *IEEE 83rd Vehicular Technology Conference: VTC2016-Spring*, Nanjing, China, 2016, pp. 1–5.

[12] C. Ren, J. Galy, E. Chaumette, F. Vincent, P. Larzabal, and A. Renaux, "Recursive hybrid Cramér-Rao bound for discrete-time Markovian dynamic systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1543–1547, Oct 2015.

[13] Y. Noam and H. Messer, "Notes on the tightness of the hybrid Cramér-Rao lower bound," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2074–2084, June 2009.

[14] W. L. Brogan, *Modern Control Theory (3rd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1991.

[15] M. Nilsson, J. Rantakokko, M. A. Skoglund, and G. Hendeby, "Indoor positioning using multi-frequency RSS with foot-mounted INS," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2014, pp. 177–186.

[16] C. Fritsche, U. Orguner, and F. Gustafsson, "On parametric lower bounds for discrete-time filtering," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 4338–4342.

[17] P. Tichavsky, C. H. Muravchik, and A. Nehorai, "Posterior Cramér-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386–1396, May 1998.

[18] N. Bergman, "Recursive Bayesian estimation navigation and tracking applications," Ph.D. dissertation, Linköping University, 1999.

[19] L. Ljung, *System Identification (2nd Ed.): Theory for the User*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[20] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[21] T. Lang, C. Plagemann, and W. Burgard, "Adaptive non-stationary kernel regression for terrain modelling," in *The Robotics: Science and Systems Conference (RSS)*, 2007.

[22] S. Remes, M. Heinonen, and S. Kaski, "Non-Stationary Spectral Kernels," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA., 2017.

[23] A. G. Wilson, "Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes," Ph.D. dissertation, University of Cambridge, 2014.

[24] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 370–378.

[25] 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.

[26] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” in *Nonlinear Filtering Handbook*. Oxford University Press, 2011, pp. 656–704.

[27] 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.

[28] A. D. Ialongo, M. V. Wilk, and C. E. Rasmussen, “Closed-form inference and prediction in Gaussian process state-space models,” in *NIPS 2017 Time-Series Workshop*, 2017.

[29] C. Fritsche and U. Orguner, “Supplementary material for On parametric lower bounds for discrete-time filtering,” Linköping University, Department of Electrical Engineering, Linköping, Sweden, Tech. Rep. LiTH-ISY-R, 3090, 2016.

[30] M. Simandl, J. Královec, and P. Tichavský, “Filtering, predictive, and smoothing Cramér-rao bounds for discrete-time nonlinear dynamic systems,” *Automatica*, vol. 37, pp. 1703–1716, 2001.

[31] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*. New Jersey: Prentice-Hall Inc, 1993.

[32] H. L. V. Trees, *Detection, estimation, and modulation theory*. Wiley, 1968, vol. 1.

[33] 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.

[34] 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.

[35] M. Gudmundson, “Correlation model for shadow fading in mobile radio systems,” *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, Nov. 1991.

[36] Senion, “White paper: ISP 101, an introduction to indoor positioning systems.” [Online]. Available: https://senion.com/indoor-positioning-101/