A particle filter approach to approximate posterior Cramér-Rao lower bound

Aditya Tulsyan, Biao Huang, R. Bhushan Gopaluni and J. Fraser Forbes

Abstract—The posterior Cramér-Rao lower bound (PCRLB) derived in [1] provides a bound on the mean square error (MSE) obtained with any non-linear state filter. Computing the PCRLB involves solving complex, multi-dimensional expectations, which do not lend themselves to an easy analytical solution. Furthermore, any attempt to approximate it using numerical or simulation based approaches require a priori access to the true states, which may not be available, except in simulations or in carefully designed experiments. To allow recursive approximation of the PCRLB when the states are hidden or unmeasured, a new approach based on sequential Monte-Carlo (SMC) or particle filters (PF) is proposed. The approach uses SMC methods to estimate the hidden states using a sequence of the available sensor measurements. The developed method is general and can be used to approximate the PCRLB in non-linear systems with non-Gaussian state and sensor noise. The efficacy of the developed method is illustrated on two simulation examples, including a practical problem of ballistic target tracking at re-entry phase.

Index Terms—PCRLB, non-linear systems, hidden states, SMC methods, target tracking

I. INTRODUCTION

Non-linear filtering is one of the most important Bayesian inferencing methods, with several key applications in: navigation [2], guidance [3], tracking [4], fault detection [5] and fault diagnosis [6]. Within the Bayesian framework, a filtering problem aims at constructing a posterior filter density [7].

In the last few decades, several tractable algorithms based on analytical and statistical approximation of the Bayesian filtering (e.g., extended Kalman filter (EKF) and unscented Kalman filter (UKF)) have been developed to allow tracking in non-linear SSMs [8]. Although filters, such as EKF and UKF are efficient in tracking, their performance is often limited or non-linear SSMs [8]. Although filters, such as EKF and UKF are efficient in tracking, their performance is often limited or

...
and $Pr_f = 0$ [27]; and cluttered environments ($Pr_d \leq 1$ and $Pr_f \geq 0$) [27]. However, unlike the bound formulation given in [1], the modified versions of the lower bound are mostly for a special class of non-linear SSMs with additive Gaussian state and sensor noise.

Notwithstanding a recursive procedure to compute the PCRLB in [1], obtaining a closed form solution to it is non-trivial. This is due to the involved complex, multi-dimensional expectations with respect to the states and measurements, which do not lend themselves to an easy analytical solution, except in linear systems [12], where the Kalman filter (KF) provides an exact solution to the PCRLB.

Several attempts have been made in the past to address the aforementioned issues. First, several authors considered approximating the PCRLB for systems with: (i) linear state dynamics with additive Gaussian noise and non-linear measurement model [12], [23]; (ii) linear and non-linear SSMs with additive Gaussian state and sensor noise [9], [29]; and (iii) linear SSMs with unknown measurement uncertainty [30]. The special sub-class of non-linear SSMs with additive Gaussian noise allows reduction of the complex, multi-dimensional expectations to a lower dimension, which are relatively easier to approximate.

II. MOTIVATION AND CONTRIBUTIONS

To obtain a reasonable approximation to the PCRLB for general non-linear SSMs, several authors have considered using simulation based techniques, such as the Monte Carlo (MC) method. Although a MC method makes the lower bound computations off-line, nevertheless, it is a popular approach, since for many real-time applications in tracking and navigation, the design, selection and performance evaluation of different filtering algorithms are mostly done a priori or off-line. Furthermore, availability of huge amount of historical test-data, makes MC method a viable option. An MC based bound approximation have appeared for several systems with: target generated measurements [13], [23]; measurement origin uncertainty [25]; cluttered environments [27], [31]; and Markovian models [32], [33]. Although MC methods can be effectively used to approximate the involved expectations, with respect to the states and measurements, it requires an ensemble of the true states and measurements. While the sensor readings may be available from the historical test-data, the true states may not be available, except in simulations or in carefully designed experiments [34].

To avoid having to use the true states, [34] proposed an EKF and UKF based method to compute the PCRLB formulation in [1]. To approximate the bound, [34] first assumes the densities associated with the expectations to be Gaussian, and then uses an EKF and UKF to approximate the Gaussian densities using an estimate of the mean and covariance. Even though the method proposed in [34] is fast, since it only works with the first two statistical moments, there are several performance and applicability related issues with this numerical approach, such as: (i) relies on the linearisation of the underlying non-linear dynamics around the state estimates, which not only results in additional numerical errors, but also introduces bias in the PCRLB approximation; (ii) the method is applicable only for non-linear SSMs with additive Gaussian state and sensor noise; (iii) convergence of the numerical solution to the theoretical lower bound is not guaranteed; (iv) provides limited control for improving the quality of the resulting numerical solution; and (v) it involves long and tedious calculations of the first two moments of the assumed Gaussian densities.

Recently, [35] derived a conditional lower bound for general non-linear SSMs, and used an SMC based method to approximate it in absence of the true states. Unlike the unconditional PCRLB in [1], the conditional PCRLB can be computed in real-time; however, as shown in [35], the bound in less optimistic (or higher) compared to the unconditional PCRLB. This limits its use to applications, where real-time bound computation is far more important than obtaining a tighter limit on the tracking performance. However, in applications, such as filter design and selection, where the primary focus is on devising an efficient filtering strategy, the PCRLB in [1] provides an optimistic measure of the filter performance.

To the authors’ best knowledge, there are no known numerical method to approximate the unconditional PCRLB in [1], when the true states are unavailable.

The following are the main contributions in this paper: (i) an SMC based method is developed to numerically approximate the unconditional PCRLB in [1], for a general stochastic non-linear SSMs operating with $Pr_d = 1$ and $Pr_f = 0$. The expectations defined originally with respect to the true states and measurements are reformulated to accommodate use of the available sensor readings. This is done by first conditioning the distribution of the true states over the sensor readings, and then using an SMC method to approximate it. (ii) Based on the above developments, a numerical method to compute the lower bound for a class of discrete-time, non-linear SSMs with additive Gaussian state and sensor noise is derived. This is required, since several practical problems, especially in tracking, navigation and sensor management, are often modelled as non-linear SSMs, with additive Gaussian noise. (iii) Convergence results for the SMC based PCRLB approximation is also provided. (iii) The quality of the SMC based PCRLB approximation is illustrated on two examples, which include a uni-variate, non-stationary growth model and a practical problem of ballistic target tracking at re-entry phase.

The proposed simulation based method is an off-line method, which can be used to deliver an efficient numerical approximation to the lower bound in [1], based on the sensor readings alone. Compared to the EKF and UKF based PCRLB approximation method derived in [34], the proposed SMC based method: (i) is far more general as it can approximate the PCRLB for a larger class of discrete-time, non-linear SSMs with possibly non-Gaussian state and sensor noise; (ii) avoids numerical errors arising due to the use of dynamics linearisation methods; and (iii) provides a far greater control over the quality of the resulting approximation. Moreover, several theoretical results exist for the SMC methods, which can be used to suggest convergence of the SMC based PCRLB approximation to the actual lower bound. All these features of the proposed method are either validated theoretically or
also the parameters operating with probability of false alarm be known a priori. In this paper, we consider a model for a class of general stochastic non-linear systems.

**Model 3.1:** Consider the following discrete-time, stochastic non-linear SSM

\[
X_{t+1} = f_t(X_t, u_t, \theta, V_t), \quad (1a) \\
Y_t = g_t(X_t, u_t, \theta, W_t), \quad (1b)
\]

where: \(X_t \in \mathcal{X} \subseteq \mathbb{R}^n\) and \(Y_t \in \mathcal{Y} \subseteq \mathbb{R}^m\) are the state variables and sensor measurements, respectively; \(u_t \in \mathcal{U} \subseteq \mathbb{R}^p\) is input variables and \(\theta \in \Theta \subseteq \mathbb{R}^r\) are the model parameters. Also: the state and sensor noise are represented as \(V_t \in \mathbb{R}^n\) and \(W_t \in \mathbb{R}^m\), respectively. \(f_t(\cdot)\) is an n-dimensional state mapping function and \(g_t(\cdot)\) is a m-dimensional measurement mapping function, where each being possibly non-linear in its arguments.

Model 3.1 represents one of the most general classes of discrete-time, stochastic non-linear SSSs. For notational simplicity, explicit dependence on \(u_t \in \mathcal{U}\) and \(\theta \in \Theta\) are not shown in the rest of this article; however, all the derivations that appear in this paper hold with \(u_t\) and \(\theta\) included. Assumptions on Model 3.1 are discussed next.

**Assumption 3.2:** The state and sensor dynamics are defined as \(f_t : \mathcal{X} \times \mathbb{R}^n \rightarrow \mathbb{R}^n\) and \(g_t : \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}^m\), respectively, are at least twice differentiable with respect to \(X_t \in \mathcal{X}\). Also: the parameters \(\theta \in \Theta\) and inputs \(u_t \in \mathcal{U}\) are assumed to be known a priori.

**Assumption 3.3:** Sensor measurements are target-originated, operating with probability of false alarm \(Pr_f = 0\) and probability of detection \(Pr_d = 1\). The target states \(X_t \in \mathcal{X}\) are hidden Markov process, observed only through the measurement process \(Y_t \in \mathcal{Y}\).

**Assumption 3.4:** \(V_t, W_t\) and \(X_0\) are mutually independent sequences of independent random variables described by the probability density functions (pdfs) \(p(v_t), p(w_t)\) and \(p(x_0)\), respectively. These pdfs are known in their classes (e.g., Gaussian; uniform) and are parametrized by a known and finite number of moments (e.g., mean; variance).

**Assumption 3.5:** For a random realization \((x_{t+1}, x_t, v_t) \in \mathcal{X} \times \mathcal{X} \times \mathbb{R}^n\) and \((y_{t+1}, x_t, w_t) \in \mathcal{Y} \times \mathcal{X} \times \mathbb{R}^m\) satisfying Model 3.1 \(\nabla_v f_t^T(x_t, v_t)\) and \(\nabla_{u_t} g_t^T(x_t, w_t)\) have rank \(n\) and \(m\), such that using implicit function theorem, \(p(x_{t+1} | x_t) = p(V_t = f_t(x_t, x_{t+1}))\) and \(p(y_{t+1} | x_t) = p(W_t = g_t(x_t, w_t))\) do not involve Dirac delta functions.

### A. Posteriori Cramér-Rao lower bound

The conventional CRLB provides a lower bound on the MSE of any ML based estimator. An analogous extension of the CRLB to the class of Bayesian estimators was derived by [10], and is referred to as the PCRLB inequality. Extension of the PCRLB to non-linear tracking was provided by [1], and is given next.

**Lemma 3.6:** Let \(\{Y_{1:t}\}_{t \in \mathbb{N}}\) be a sequence from Model 3.1 then MSE of any tracking filter at \(t \in \mathbb{N}\) is bounded from below by the following matrix inequality

\[
P_{t|t} \triangleq \mathbb{E}_{p(x_{0:t+1}, Y_{1:t})}[ (X_{t} - \hat{X}_{t|t})(X_{t} - \hat{X}_{t|t})^T ] \geq J_{t}^{-1},
\]

where: \(P_{t|t}\) is a \(n \times n\) matrix of MSE; \(\hat{X}_{t|t} \triangleq \mathbb{E}_{p(x_{0:t+1}, Y_{1:t})}[ Y_{t} ]\) is a point estimate of \(X_t \in \mathcal{X}\) at time \(t \in \mathbb{N}\), given the measurement sequence \(\{Y_{1:t} = y_{1:t}\} \triangleq \{y_1, \ldots, y_t\}\); \(J_t\) is a \(n \times n\) PCRLB matrix; \(p(x_{0:t}, y_{1:t})\) is a joint probability density of the states and measurements up until time \(t \in \mathbb{N}\); the superscript \((\cdot)^T\) is the transpose operation; and \(\mathbb{E}_{p(\cdot)}[\cdot]\) is the expectation operator with respect to the pdf \(p(\cdot)\).

**Proof:** See [10] for a detailed proof. □

Inequality 3 implies that \(P_{t|t} - J_t^{-1} \succ 0\) is a positive semi-definite matrix for all \(\hat{X}_{t|t} \in \mathbb{R}^n\) and \(t \in \mathbb{N}\). (2) can also be written in terms of a scalar MSE (SMSE) as

\[
P_{t|t}^S \triangleq \mathbb{E}_{p(x_{0:t}, Y_{1:t})}[ ||X_{t} - \hat{X}_{t|t}||^2 ] \geq \text{Tr}[J_{t}^{-1}],
\]

where \(\text{Tr}[\cdot]\) is the trace operator, and \(\| \cdot \|\) is a 2-norm.

**Lemma 3.7:** For a system represented by Model 3.1 and operating under Assumptions 3.2 through 3.5, the PFIM in Lemma 3.6 can be recursively computed as [1]. [9]

\[
J_{t+1} = D_{t}^{22} - [D_{t}^{12}]^T (J_t + D_{t}^{11})^{-1} D_{t}^{12},
\]

where:

\[
D_{t}^{11} = \mathbb{E}_{p(x_{0:t}, Y_{1:t})}[ -\nabla^2_{X_t} \log p(X_{t+1} | X_t)];
\]

\[
D_{t}^{12} = \mathbb{E}_{p(x_{0:t}, Y_{1:t})}[ -\nabla^2_{X_t} \log p(Y_{1:t} | X_t)];
\]

\[
D_{t}^{22} = \mathbb{E}_{p(x_{0:t}, Y_{1:t})}[ -\nabla^2_{Y_{t+1}} \log p(Y_{t+1} | X_{t+1})] - \nabla^2_{X_{t+1}} \log p(Y_{t+1} | X_{t+1})];
\]

and: \(\Delta\) is a Laplacian operator such that \(\nabla^2_{X} \triangleq \nabla \nabla^T_{X}\) with \(\nabla_{X} \triangleq \left[ \frac{\partial}{\partial x} \right] \) being a gradient operator, evaluated at the true states. Also, \(J_0 = \mathbb{E}_{p(x_0)}[ -\nabla^2_{X_0} \log p(X_0)]\).

**Proof:** See [1] for a complete proof. □

For Model 3.1 obtaining a closed-form solution to the PFIM or PCRLB is non-trivial. This is due to the complex integrals involved in 3.5, which do not lend themselves to an easy analytical solution. The main problem addressed in this paper is discussed next.

**Problem 3.8:** Compute a numerical solution to the PCRLB given in Lemma 3.6 for systems represented by Model 3.1 and operating under Assumptions 3.2 through 3.5. Use of simulation based methods in addressing Problem 3.8 is discussed next.

### IV. Approximating PCRLB

MC method is a popular approach, which can be used to approximate the PCRLB; however, as discussed in Section II MC method requires an ensemble of true states and sensor measurements. While sensor readings may be available from the historical test-data, the true states may not be available in practice. To allow the use of sensor readings in approximating the PCRLB, this paper reformulates the integrals in 3.5 as given below.
Proposition 4.1: The complex, multi-dimensional expectations in (5), with respect to the density \( p(x_{0:t+1}, y_{1:t+1}) \) can be reformulated, and written as follows:

\[I_{t}^{11} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t+1} \log p(X_{t+1}|X_{t})];\]  
\[I_{t}^{22} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t+1} \log p(Y_{t+1}|X_{t})];\]  
\[I_{t}^{22,a} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t+1} \log p(X_{t+1}|Y_{t})];\]  
\[I_{t}^{22,b} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t+1} \log p(Y_{t+1}|X_{t})].\]  

where:

\[D_{t}^{11} = \mathbb{E}_{p(Y_{t+1}|Y_{t})}[I_{t}^{11}];\]  
\[D_{t}^{22} = \mathbb{E}_{p(Y_{t+1}|Y_{t})}[I_{t}^{22,a} + I_{t}^{22,b}].\]

Proof: The proof is based on decomposition of the pdf \( p(x_{0:t+1}, y_{1:t+1}) \) in (5), using the probability condition \( p(x_{0:t+1}, y_{1:t+1}) = p(y_{1:t+1})p(x_{0:t+1}|y_{1:t+1}). \)

Remark 4.2: In Proposition 4.1, the integrals are with respect to \( p(y_{1:t+1}) \) and \( p(x_{0:t+1}|y_{1:t+1}). \) The advantage of representing (5) as (6) is evident: using historical test-data, expectations with respect to \( p(y_{1:t+1}) \) can be approximated using MC, while that defined with respect to \( p(x_{0:t+1}|y_{1:t+1}) \) can be approximated using an SMC method.

A. SMC based PCRLB approximation

It is not our aim here to review SMC methods in details, but to simply highlight their role in approximating the multi-dimensional integrals in Proposition 4.1. For a detailed exposition on SMC methods, see [7, 11]. The essential idea behind SMC methods is to generate a large set of random particles (samples) from the target pdf, with respect to which the integrals are defined. The target pdf of interest in Proposition 4.1 is \( p(x_{0:t}, y_{1:t}). \) Using SMC methods, the target distribution, defined as \( \hat{p}(dx_{0:t+1}|y_{1:t+1}) \triangleq p(x_{0:t+1}|y_{1:t+1})dx_{0:t+1} \) can be approximated as given below:

\[\hat{p}(dx_{0:t+1}|y_{1:t+1}) = \sum_{i=1}^{N} W_{0:t+1|t+1}^{i} \delta_{X_{0:t+1|t+1}}^{i}(dx_{0:t+1}),\]  

where: \( \hat{p}(dx_{0:t+1}|y_{1:t+1}) \) is an \( N \)-particle SMC approximation of the target distribution \( p(dx_{0:t+1}|y_{1:t+1}) \) and \( \{X_{0:t+1|t+1}^{i}\}_{i=1}^{N} \) are the \( N \) pairs of particle realizations and their associated weights distributed according to \( p(x_{0:t+1}|y_{1:t+1}), \) such that \( \sum_{i=1}^{N} W_{0:t+1}^{i} = 1. \) Using (7), an SMC approximation of (6a), for example, can be computed as

\[\hat{I}_{t}^{11} = \sum_{i=1}^{N} W_{0:t+1|t+1}^{i}[-\Delta X_{t} \log p(X_{t+1|t+1}^{i}|X_{t}).\]  

Theorem 4.3: For any bounded test function \( \phi_{t} : X^{t+1} \rightarrow \mathbb{R} \), there exists \( C_{t,p} < \infty \) such that for any \( p > 0, N \geq 1 \) and \( t \geq 1 \), the following inequality holds

\[E \left[ \left| \int_{X^{t+1}} \phi_{t}(x_{0:t}) \epsilon_{L}(dx_{0:t}|y_{1:t}) \right|^{p} \right]^{1/p} \leq C_{t,p} \delta_{t} N^{1/2}.\]  

where \( \epsilon_{L}(dx_{0:t}|y_{1:t}) = \hat{p}(dx_{0:t}|y_{1:t}) - p(dx_{0:t}|y_{1:t}) \) is the \( N \)-particle approximation error, \( \delta_{t} = \sup_{x_{0:t} \in X^{t+1}}|\phi_{t}(x_{0:t})| \), and the expectation is with respect to the particle realizations.

Proof: See Theorem 2 in [38] for a detailed proof.

Remark 4.4: The result in Theorem 4.3 is weak, since \( C_{t,p} \in \mathbb{R} \) being a function of \( t \in \mathbb{N} \), grows exponentially/polyonomially with time [39]. To guarantee a fixed precision of the approximation in (8), \( N \) has to increase with \( t. \) The result in Theorem 4.3 is not surprising, since (7) requires sampling from the pdf \( p(x_{0:t+1}|y_{1:t}), \) whose dimension increases as \( n(t+1) \). In literature Theorem 4.3 is referred to as the sample path degeneracy problem. This is a fundamental limitation of SMC methods; wherein, for \( N \in \mathbb{N} \), the quality of the approximation of \( \hat{p}(dx_{0:t+1}|y_{1:t}) \) deteriorates with time.

The motivation to use SMC methods to approximate the complex, multi-dimensional integrals in Proposition 4.1 is based on the fact that encouraging results can be obtained under the exponential forgetting assumption on Model 3.1 Since \( \theta \in \Theta \) is assumed to be known (see Assumption 3.2), the forgetting property in Model 3.1 holds. With the forgetting property, it is possible to establish results of the form given in the next theorem.

Theorem 4.5: For an integer \( L > 0 \), and any bounded test function \( \phi_{L} : X^{L} \rightarrow \mathbb{R} \), there exists \( D_{L,p} < \infty \), such that for any \( p > 0, N \geq 1 \) and \( t \geq 1 \), the following inequality holds

\[E \left[ \left| \int_{X^{t+1}} \phi_{L}(x_{t-L+1:t}) \epsilon_{L}(dx_{t-L+1:t}|y_{1:t}) \right|^{p} \right]^{1/p} \leq D_{L,p} \delta_{t} N^{1/2}.\]  

where \( \epsilon_{L}(dx_{t-L+1:t}|y_{1:t}) = \int_{X^{t-L+1}} \epsilon_{L}(dx_{0:t}|y_{1:t}). \)

Proof: See Theorem 2 in [38] for a detailed proof.

Remark 4.6: Since \( D_{L,p} \in \mathbb{R} \) is independent of \( t \in \mathbb{N} \), Theorem 4.5 suggests that an SMC based approximation of the most recent marginal posterior pdf \( p(x_{t-L+1:t}|y_{1:t}) \), over a fixed horizon \( L > 0 \) does not result in the error accumulation.

For our purposes, to make the SMC based PCRLB approximation effective, the dimension of the integrals in Proposition 4.1 needs to be reduced. An SMC based approximation of the PCRLB over a reduced dimensional state-space is discussed next.

Lemma 4.7: For a system represented by Model 3.1 using the Markov property of the target states in Assumptions 3.3 Proposition 4.1 can be written as follows:

\[I_{t}^{11} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t} \log p(X_{t+1}|X_{t})];\]  
\[I_{t}^{22} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t} \log p(Y_{t+1}|X_{t})];\]  
\[I_{t}^{22,a} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t} \log p(X_{t+1}|Y_{t})];\]  
\[I_{t}^{22,b} = \mathbb{E}_{p(x_{0:t+1}|Y_{1:t+1})}[-\Delta X_{t} \log p(Y_{t+1}|X_{t})].\]

Proof: The proof is based on a straightforward use of the definition of expectation and Markov property of Model 3.1.
For example, the integrals in (6a) can be written as

\[ I_t^{(1)} = \int_{x' \in \mathbb{R}} \left[ -\Delta x_i \log p(x_{t+1} | x_i) p(dx_{0:t+1} | y_{1:t+1}) \right], \quad (12a) \]

\[ \int_{x' \in \mathbb{R}} \left[ -\Delta x_i \log p(x_{t+1} | x_i) p(dx_{0:t+1} | y_{1:t+1}) \right], \quad (12b) \]

\[ = \mathbb{E}_{p(X_{t+1} | Y_{1:t+1})} \left[ -\Delta X_i \log p(X_{t+1} | X_i) \right], \quad (12c) \]

where \( p(dx_{0:t+1} | y_{1:t+1}) \) is independent of \( x_{0:t-1} \in \mathcal{X}^t \), and in (12c), since the integrand is independent of \( x_{0:t-1} \in \mathcal{X}^t \), it is marginalized out of the integral. Equations (11b) through (11d) can be derived based on similar arguments, which completes the proof.

*Remark 4.8:* The dimension of the expectations in (6a) through (6c) reduces from \( n(t+2) \) to \( 2n \); whereas, in (6d), it reduces from \( n(t+2) \) to \( n \) for all \( t \in \mathbb{N} \). Moreover, since expectations in Lemma 4.7 are with respect to \( p(x_{t+1} | y_{1:t+1}) \) and \( p(x_{t+1} | y_{1:t+1}) \), an SMC method can be effectively used with a finite number of particles (see Theorem 4.5).

### B. General non-linear SSMs

To approximate the multi-dimensional integrals in Lemma 4.7 for Model 3.1, a set of randomly generated samples from the target distribution \( p(dx_{t+1} | y_{1:t+1}) \) is required. First note that the target pdf \( p(x_{t+1} | y_{1:t+1}) \) can alternatively be written as given below.

**Lemma 4.9:** The target pdf \( p(x_{t+1} | y_{1:t+1}) \), with respect to which the integrals in Lemma 4.7 are defined can be decomposed, and written as

\[ p(x_{t+1} | y_{1:t+1}) = \frac{p(x_{t+1} | x_t) p(x_t | y_{1:t}) p(x_t | y_{1:t+1})}{\int_{x_t} p(x_t | x_t) p(dx_t | y_{1:t})}. \]

\[ (13) \]

**Proof:** First note that the target pdf \( p(x_{t+1} | y_{1:t+1}) \) can be written as

\[ p(x_{t+1} | y_{1:t+1}) = p(x_t | x_{t+1}, y_{1:t}, y_{1:t+1}) p(x_{t+1} | y_{1:t+1}). \]

\[ (14) \]

From the Markov property of (1), and from the Bayes’ theorem, (14) can be written as

\[ p(x_{t+1} | y_{1:t+1}) = \frac{p(y_{t+1} | x_t, x_{t+1}, y_{1:t}) p(x_t | x_{t+1}, y_{1:t}) p(x_{t+1} | y_{1:t+1})}{p(y_{t+1} | x_t, y_{1:t})}, \]

\[ (15a) \]

\[ = \frac{p(y_{t+1} | x_t, x_{t+1}, y_{1:t}) p(x_t | x_{t+1}, y_{1:t}) p(x_{t+1} | y_{1:t+1})}{p(y_{t+1} | x_{t+1}, y_{1:t})}, \]

\[ (15b) \]

\[ = p(x_t | x_{t+1}, y_{1:t}) p(x_{t+1} | y_{1:t+1}). \]

\[ (15c) \]

Applying Bayes’ theorem again in (15c) yields

\[ p(x_{t+1} | y_{1:t+1}) = \frac{p(x_{t+1} | x_t, y_{1:t}) p(x_t | y_{1:t}) p(x_{t+1} | y_{1:t+1})}{p(x_{t+1} | y_{1:t})}, \]

\[ (16a) \]

\[ = \frac{p(x_{t+1} | x_t, y_{1:t}) p(x_t | y_{1:t}) p(x_{t+1} | y_{1:t+1})}{\int_{x_t} p(x_t | x_t) p(dx_t | y_{1:t})}, \]

\[ (16b) \]

where in (16b), the Law of Total Probability is used, which completes the proof.

**Algorithm 1** SMC based posterior density approximation

**Input:** Given Model 3.1 satisfying Assumptions 3.2 through 3.5 assume a prior pdf on \( x_0 \), such that \( x_0 \sim p(x_0) \). Also, select algorithm parameter \( N \).

**Output:** Recursive SMC approximation of the posterior \( p(dx_t | y_{1:t}) \) for all \( t \in \mathbb{N} \).

1: Generate \( N \) independent and identically distributed particles \( \{X_{i,0} \}_{i=1}^N \sim p(x_0) \) and set the associated weights to \( \{W_{i,0} \}_{i=1}^N \). Set \( t \leftarrow 1 \).

2: Sample \( \{X_{i,t-1} \}_{i=1}^N \) using (18). Set \( \{W_{i,t-1} \}_{i=1}^N \sim p(x_t | y_{1:t-1}) \).

3: while \( t \in \mathbb{N} \) do

4: Use \( \{Y_t = y_t \} \) and compute the importance weights \( \{W_{i,t} \}_{i=1}^N \) using (18).

5: Resample the particle set \( \{X^j_t \}_{j=1}^N \) with replacement from \( \{X_{i,t-1} \}_{i=1}^N \), such that

\[ Pr(X^j_t = X^i_{t-1}) = W^i_{t-1}, \]  

\[ (19) \]

where \( Pr(\cdot) \) is a probability measure. Set \( \{W^j_t \}_{i=1}^N \) where

6: Sample \( \{X^i_{t+1} \}_{i=1}^N \sim p(x_{t+1} | y_{1:t}) \) using (17). Set \( \{W^i_{t+1} \}_{i=1}^N \sim p(x_{t+1} | y_{1:t}) \).

7: Set \( t \leftarrow t + 1 \).

8: end while

**Remark 4.10:** The procedure for generating random particles from densities, such as the uniform or Gaussian, is well described in literature; however, due to the multi-variate, and non-Gaussian nature of the target pdf, generating random particles from \( p(x_{t+1} | y_{1:t+1}) \) is a non-trivial problem. An alternative idea is to employ an importance sampling function (ISF), from which random particles are easier to generate [7].

In this paper, the product of two pdfs in (13) is selected as the ISF, such that

\[ q(x_{t+1} | y_{1:t+1}) \triangleq p(x_t | y_{1:t}) p(x_{t+1} | y_{1:t+1}), \]

\[ (17) \]

where \( q(x_{t+1} | y_{1:t+1}) \) is a non-negative ISF on \( \mathcal{X}^2 \), such that \( \text{supp} q(x_{t+1} | y_{1:t+1}) \supseteq \text{supp} p(x_{t+1} | y_{1:t+1}) \). Choice of an ISF similar to (17) was also employed in [40, 41] to develop a particle smoothing algorithm for discrete-time, non-linear SSMs. Thus to be able to generate random samples from (17), samples from the two posteriors \( p(x_t | y_{1:t}) \) and \( p(x_{t+1} | y_{1:t+1}) \) need to be generated first. Again, using the principles of ISF, particles from the posterior pdf can be generated using any advanced SMC methods (e.g., ASIR [42], resample-move algorithm [43], block sampling strategy [44]) or for example, using the method in [41, 45]. The method described in [41, 45] is outlined in Algorithm 1. It is important to note that in importance sampling, degeneracy is a common problem; wherein, after a few time instances, the density of the weights in (18) become skewed. The resampling step in
is crucial in limiting the effects of degeneracy. Finally using Algorithm \ref{algo:example}, the particle representation of \( p(dx_t | y_{1:t}) \) and \( p(dx_{t+1} | y_{1:t+1}) \) are given by

\[ \hat{p}(dx_t | y_{1:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t,i}}^t (dx_t), \quad (20a) \]
\[ \hat{p}(dx_{t+1} | y_{1:t+1}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{t+1,i}}^{t+1} (dx_{t+1}). \quad (20b) \]

Here \( \{X_{t,i}^t\}_{i=1}^{N} \sim \hat{p}(x_t | y_{1:t}) \) and \( \{X_{t+1,i}^{t+1}\}_{i=1}^{N} \sim \hat{p}(x_{t+1} | y_{1:t+1}) \) are the \( N \) pairs of resampled i.i.d. samples from \( \hat{p}(x_t | y_{1:t}) \) and \( \hat{p}(x_{t+1} | y_{1:t+1}) \), respectively.

**Remark 4.11:** Uniform convergence in time of (20) has been established by \cite{37, 46}. Although these results rely on strong mixing assumptions of Model \ref{model:example}, uniform convergence has been observed in numerical studies for a wide class of non-linear time-series models, where the mixing assumptions are not satisfied.

Substituting (20) into (17), yields an SMC approximation of the ISF, i.e.,

\[ \tilde{q}(dx_{t:t+1} | y_{1:t+1}) = \frac{1}{N^2} \sum_{j=1}^{N} \sum_{i=1}^{N} \delta_{X_{t,i}^j,X_{t+1,i}^{j+1}}^j (dx_{t:t+1}), \quad (21) \]

where \( \tilde{q}(dx_{t:t+1} | y_{1:t+1}) \) is an \( N^2 \)-particle SMC approximation of the ISF distribution \( q(dx_{t:t+1} | y_{1:t+1}) \) and \( \{X_{t,i}^j,X_{t+1,i}^{j+1}\}_{i=1,j=1}^{N,N} \sim \tilde{q}(x_{t:t+1} | y_{1:t+1}) \) are particles from the ISF.

**Lemma 4.12:** An SMC approximation of the target distribution \( p(dx_{t:t+1} | y_{1:t+1}) \) can be computed using the SMC approximation of \( q(dx_{t:t+1} | y_{1:t+1}) \) given in (21), such that

\[ \hat{p}(dx_{t:t+1} | y_{1:t+1}) = \sum_{i=1}^{N} W_{i,t,t+1}^i \delta_{X_{t,i}^j,X_{t+1,i}^{j+1}}^j (dx_{t:t+1}), \quad (22) \]

where:

\[ W_{i,t,t+1}^i \triangleq \frac{c_{i,t,t+1}^i}{\sum_{j=1}^{N} c_{j,t,t+1}^j}; \quad (23a) \]
\[ c_{i,t,t+1}^i \triangleq \frac{p(X_{t+1,i}^{j+1} | y_{1:t+1})}{N \sum_{m=1}^{N} p(X_{t+1,i}^{j+1} | y_{1:t} X_{t}^m)}. \quad (23b) \]

and \( \hat{p}(dx_{t:t+1} | y_{1:t+1}) \) is an SMC approximation of the target distribution \( p(dx_{t:t+1} | y_{1:t+1}) \).

**Proof:** Substituting (21) into (13) followed by several algebraic manipulations yields an SMC approximation of \( p(dx_{t:t+1} | y_{1:t+1}) \), denoted by \( \tilde{p}(dx_{t:t+1} | y_{1:t+1}) \), such that

\[ \tilde{p}(dx_{t:t+1} | y_{1:t+1}) = \frac{\int_X p(x_{t+1} | x_t) \hat{q}(dx_{t:t+1} | y_{1:t+1})}{\int_X p(x_{t+1} | x_t) \hat{p}(dx_{t+1} | y_{1:t+1})}, \quad (24a) \]
\[ = \frac{N \int_X p(x_{t+1} | x_t) \sum_{i=1}^{N} \sum_{i=1}^{N} \delta_{X_{t,i}^j,X_{t+1,i}^{j+1}}^j (dx_{t:t+1})}{N^2 \int_X p(x_{t+1} | x_t) \sum_{m=1}^{N} \delta_{X_{t,i}^j} (dx_i)}, \quad (24b) \]

Finally using Algorithm \ref{algo:example} and \( N \) is the total number of i.i.d. measurement sequences obtained from the historical test-data. Note that the approximation in (29) is possible only under Assumption \ref{assumption:example}, however, in general, estimating the marginalized likelihood function \( p(y_{1:t+1}) \) is non-trivial \cite{39}.
Finally, an SMC approximation of the PCRLB for systems represented by Model \( \mathcal{M}_1 \) and operating under Assumptions \( \mathcal{A}_2 \) through \( \mathcal{A}_5 \) is summarized in the next lemma.

**Lemma 4.14**: Let a general non-linear system be represented by Model \( \mathcal{M}_1 \) such that it satisfies Assumption \( \mathcal{A}_2 \) through \( \mathcal{A}_5 \). Let \( \{Y_{1:t} = y_{1:t}^M\}_{j=1}^M \) be \( M \in \mathbb{N} \) i.i.d. measurement sequences generated from Model \( \mathcal{M}_1 \) then the matrices \( \mathcal{B}_j \) through \( \mathcal{B}_k \) in Lemma 3.7 can be recursively approximated as follows:

\[
\tilde{D}_t^{11} = -\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [\Delta X_{ti}^j \log p(X_t(i,t+1) | X_{t+1}(i))];
\]

\[(30a)\]

\[
\tilde{D}_t^{12} = -\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [\Delta X_{ti}^{j+1} \log p(X_t(i,t+1) | X_{t+1}(i)));
\]

\[(30b)\]

\[
\tilde{D}_t^{22} = -\frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [\Delta X_{ti}^{j+1} \log p(X_t(i,t+1) | X_{t+1}(i)));
\]

\[
\quad + \Delta X_{ti}^{j+1} \log p(Y_{j+1}^t(i,t+1) | X_{t+1}(i));
\]

\[(30c)\]

and \( \{X_{ti}^{j,t+1} \}_{i=1}^N \sim p(x_{ti+1} | y_{j+1}^t) \) is a set of \( N \) resampled particles from \( \mathcal{B}_j \), distributed according to \( p(x_{t+1} | y_{j+1}^t) \) for all \( \{Y_{1:t} = y_{1:t}^M\}_{j=1}^M \).

**Proof**: For a measurement sequence \( \{Y_{1:t} = y_{1:t}^M\} \), an SMC approximation of the target distribution in \( \mathcal{B}_j \) can be written as

\[
\tilde{p}(dx_{t+1} | y_{j+1}^t) = \frac{1}{N} \sum_{i=1}^N \delta_{X_{ti}^{j,t+1}}(dx_{t+1}),
\]

\[(31)\]

where \( X_{ti}^{j,t+1} \sim p(x_{t+1} | y_{j+1}^t) \) are resampled particles. Substituting \( (31) \) into Lemma 3.7 an SMC approximation of \( I_t \) through \( I_d \) can be obtained as follows:

\[
\tilde{I}_t^{11} = \frac{1}{N} \sum_{i=1}^N -\Delta X_{ti}^j \log p(X_t(i,t+1) | X_{t+1}(i));
\]

\[(32a)\]

\[
\tilde{I}_t^{12} = \frac{1}{N} \sum_{i=1}^N -\Delta X_{ti}^{j+1} \log p(X_t(i,t+1) | X_{t+1}(i));
\]

\[(32b)\]

\[
\tilde{I}_t^{22,a} = \frac{1}{N} \sum_{i=1}^N -\Delta X_{ti}^{j+1} \log p(X_t(i,t+1) | X_{t+1}(i));
\]

\[(32c)\]

\[
\tilde{I}_t^{22,b} = \frac{1}{N} \sum_{i=1}^N -\Delta X_{ti}^{j+1} \log p(Y_{j+1}^t(i,t+1) | X_{t+1}(i)));
\]

\[(32d)\]

where \( \tilde{I}_t \) is an SMC approximation of \( I_t \). Substituting \( (32) \) and \( (29) \) into \( (5a) \) through \( (5c) \) yields \( (30a) \) through \( (30c) \), which completes the proof. \( \square \)

Lemma 4.14 gives an SMC based numerical method to approximate the complex, multi-dimensional integrals in Lemma 3.7. Note that since Lemma 4.14 is valid for a general non-linear SSMs, the derivatives of the logarithms of the pdfs in \( (30a) \) through \( (30c) \) are left in its original form, but can be computed for a given system.

Building on the developments in this section, an SMC approximation of the PCRLB for a class of non-linear SSMs with additive Gaussian noise is presented next.

### C. Non-linear SSMs with additive Gaussian noise

Many practical applications in tracking (e.g., ballistic target tracking [13], bearings-only tracking [43], range-only tracking [49], multi-sensor resource deployment [17] and other navigation problems [59]) can be described by non-linear SSMs with additive Gaussian noise. Since the class of practical problems with additive Gaussian noise is extensive, especially in tracking, navigation and sensor management, an SMC based numerical method for approximating the PCRLB for such class of non-linear systems is presented.

**Model 4.15**: Consider the class of non-linear SSMs with additive Gaussian noise

\[
X_{t+1} = f_t(X_t) + V_t,
\]

\[(33a)\]

\[
Y_t = g_t(X_t) + W_t,
\]

\[(33b)\]

where \( V_t \in \mathbb{R}^n \) and \( W_t \in \mathbb{R}^m \) are mutually independent sequences from the Gaussian distribution, such that \( V_t \sim \mathcal{N}(v_t | 0, Q_t) \) and \( W_t \sim \mathcal{N}(w_t | 0, R_t) \).

Note that Model 4.15 can also be represented as

\[
\log[p(X_{t+1} | X_t)] = c_t - \frac{1}{2} [X_{t+1} - f_t(X_t)]^T Q_t^{-1} [X_{t+1} - f_t(X_t)],
\]

\[(34a)\]

\[
\log[p(Y_{t+1} | X_{t+1})] = c_{t+1} - \frac{1}{2} [Y_{t+1} - g_t(X_{t+1})]^T R_t^{-1} [Y_{t+1} - g_t(X_{t+1})],
\]

\[(34b)\]

where \( c_t \in \mathbb{R}_+ \) and \( c_{t+1} \in \mathbb{R}_+ \) are normalizing constant and \( \mathbb{R}_+ := [0, \infty) \).

**Result 4.16**: The first and second order partial derivative of \( (34a) \) is given by

\[
\nabla_{X_t} \log[p(X_{t+1} | X_t)] = \nabla_{X_t} f_t^T(X_t) Q_t^{-1} [X_{t+1} - f_t(X_t)],
\]

\[(35a)\]

\[
\Delta_{X_t}^2 \log[p(X_{t+1} | X_t)] = \nabla_{X_t} f_t^T(X_t) \nabla_{X_t} f_t(X_t) + \Delta_{X_t}^2 f_t^T(X_t) X_t^{-1} \nabla_{X_t} f_t(X_t),
\]

\[(35b)\]

and the first with respect to \( X_{t+1} \in \mathcal{X} \) and the second with respect to \( X_t \in \mathcal{X} \) is given by

\[
\Delta_{X_{t+1}} \log[p(X_{t+1} | X_t)] = [\nabla_{X_t} f_t^T(X_t) Q_t]^{-1},
\]

\[(35c)\]

where: \( \Delta_{X_t} X_t = Q_t^{-1} n_x \times n_x \); \( \Psi_{X_t} = \left[ X_{t+1} - f_t(X_t) \right] I_{n_x \times n_x} \); \( I_{n_x \times n_x} \) and \( I_2 \times n_x \) are \( n_x^2 \times n_x \) and \( n_x^2 \times n_x \) identity matrix, respectively. Also: \( \nabla_{X_t} f_t^T(X_t) \) and \( \Delta_{X_t}^2 f_t^T(X_t) \) are

\[
[\nabla_{X_t} f_t^T(X_t)] \triangleq [\nabla_{X_t} f_t^{(1)}(X_t), \ldots, \nabla_{X_t} f_t^{(n)}(X_t)]_{n_x \times n},
\]

\[(36a)\]

\[
[\Delta_{X_t}^2 f_t^T(X_t)] \triangleq [\Delta_{X_t}^2 f_t^{(1)}(X_t), \ldots, \Delta_{X_t}^2 f_t^{(n)}(X_t)]_{n_x \times n},
\]

\[(36b)\]

where \( f_t(X_t) \triangleq [f_t^{(1)}(X_t), \ldots, f_t^{(n)}(X_t)]^T \) is a \( n \times 1 \) vector valued function in \( 33a \).

**Result 4.17**: The second order partial derivative of \( (34a) \) is given by

\[
\Delta_{X_{t+1}}^2 \log[p(X_{t+1} | X_t)] = -Q_t^{-1}
\]

\[(37a)\]

\[
\Delta_{X_{t+1}}^2 \log[p(Y_{t+1} | X_{t+1})] = \Delta_{X_t}^2 g_t^T(X_{t+1}) A_{g_t}^{-1} \Psi_{Y_{t+1}} - [\nabla_{X_t} g_t^T(X_{t+1})] R_t^{-1} [\nabla_{X_t} g_t(X_{t+1})],
\]

\[(37b)\]
where: \( A_{X_{t+1}}^{-1} = R_{X_t, X_{t+1}}^{-1} I_{n^2 \times n^2}; \)
\( \Psi_{Y_{t+1}} = [Y_{t+1} - g_{t+1}(X_{t+1})] I_{n^2 \times n^2}; \)
and \( I_{n^2 \times n^2} \) are \( n^2 \times n^2 \) and \( n^2 \times n \) identity matrices. Also: \(|\nabla X_{t+1} g_{t+1}(X_{t+1})|\)
and \( \Delta X_{t+1} g_{t+1}(X_{t+1}) \) are
\[
[\nabla X_{t+1} g_{t+1}(X_{t+1})] = \\
= [\nabla X_{t+1} g_{t+1}^{(1)}(X_{t+1}), \ldots, \nabla X_{t+1} g_{t+1}^{(m)}(X_{t+1})]_{m \times m};
\] (38a)
\[
[\Delta X_{t+1} g_{t+1}(X_{t+1})] = \\
= [\Delta X_{t+1} g_{t+1}^{(1)}(X_{t+1}), \ldots, \Delta X_{t+1} g_{t+1}^{(m)}(X_{t+1})]_{m \times m};
\] (38b)
where \( g_{t+1}(X_{t+1}) = [g_{t+1}^{(1)}(X_{t+1}), \ldots, g_{t+1}^{(n)}(X_{t+1})]^T \) is a \( m \times 1 \) vector function in \( \mathbb{R}^m \).

**Lemma 4.18:** For a system given by Model 4.15 under Assumptions 4.2 through 4.5, the matrices \( (11a) \) through \( (11d) \) in Lemma 4.7 can be written as:
\[
I_t^{11} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (39a)
\[
I_t^{22} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (39b)
\[
I_t^{22} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (39c)
\[
I_t^{22} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (39d)

**Proof:** (39a): Substituting \( (35b) \) into \( (11a) \) yields
\[
I_t^{11} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (40a)
Finally, by noting the following two conditions
\[
E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (41)
\[
E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (42)
and substituting \((41)\) and \((42)\) into \((40b)\) yields \( (39a) \).

Substituting \( (37b) \) into \((45)\) yields
\[
I_t^{22} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (46)

**Substituting** \((37b)\) into \((45)\) yields
\[
I_t^{22} = E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (46)

**Noting the following two conditions**
\[
E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (47)
\[
E_p(X_{t+1} Y_{t+1})[\nabla_X f_t^T(X_t)] Q_t^{-1} [\nabla_X f_t(X_t)];
\] (48)
and substituting \((47)\) and \((48)\) into \((46)\) yields \( (39d) \), which completes the proof.

**Using the results of Lemma 4.18** as an SMC approximation of the PCRLB for Model 4.15 can be subsequently computed, as discussed in the next lemma.

**Lemma 4.19:** Let a stochastic non-linear system with additive Gaussian state and sensor noise be represented by Model 4.15 such that it satisfies Assumption 4.2 through 4.5. Let \( \{Y_{t+1} = y_{t+1}^i\}_{i=1}^M = M \in \mathbb{N} \) i.i.d. measurement sequences generated from Model 4.15 then \( (5a) \) through \( (5c) \) in Lemma 3.7 can be recursively approximated as follows:
\[
\tilde{D}_{t+1}^{11} = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [\nabla_X f_t^T(X_{t+i}^{j})] Q_t^{-1} [\nabla_X f_t(X_{t+i}^{j})];
\] (49a)
\[
\tilde{D}_{t+1}^{22} = \frac{1}{MN} \sum_{j=1}^M \sum_{i=1}^N [\nabla_X f_t^T(X_{t+i}^{j})] Q_t^{-1} [\nabla_X f_t(X_{t+i}^{j})];
\] (49b)

**Proof:** For \( \{Y_{t+1} = y_{t+1}^i\}\), the SMC approximation in \((28)\) can be written as
\[
\bar{p}(dx_{t+i}^j) = \frac{1}{N} \sum_{i=1}^N \delta_{X_{t+i}^j} dx_{t+i}^j;
\] (50)
where \( X_{t+i}^j \sim p(x_{t+i}^j|y_{t+1}^i) \). Substituting \((50)\) into \((39a)\) and \((39b)\) yields
\[
\tilde{I}_{t+1}^{11} = \frac{1}{N} \sum_{i=1}^N [\nabla_X f_t^T(X_{t+i}^{j})] Q_t^{-1} [\nabla_X f_t(X_{t+i}^{j})];
\] (51a)
\[
\tilde{I}_{t+1}^{22} = -\frac{1}{N} \sum_{i=1}^N [\nabla_X f_t^T(X_{t+i}^{j})] Q_t^{-1};
\] (51b)
Computing an SMC approximation of $D_t^{22}$ in (49) for Model 4.13 requires a slightly different approach. Substituting (39c) and (40d) into (49) yields

$$D_t^{22} = \mathbb{E}_p(Y_{t+1} | Q_t^{-1} + \mathbb{E}_p_Y(Y_{t+1} | \mathbb{E}_p(X_t+1 | Y_{t+1}) \times [\nabla X_{t+1} g_{t+1}^T (X_{t+1}) | R_{t+1}^{-1} \nabla X_{t+1} g_{t+1}^T (X_{t+1})]], \quad (52a)$$

$$= Q_t^{-1} + \mathbb{E}_p(Y_{t+1} | \mathbb{E}_p(X_t+1 | Y_{t+1}) \nabla X_{t+1} g_{t+1}^T (X_{t+1}) | R_{t+1}^{-1} \times [\nabla X_{t+1} g_{t+1}^T (X_{t+1})], \quad (52b)$$

where $Q_t^{-1}$ is independent of the measurement sequence. Also, $\mathbb{E}_p(Y_{t+1} | \mathbb{E}_p(Y_{t+1} | \cdot) = \mathbb{E}_p(Y_{t+1} | \cdot)$. For $\{Y_{t+1} = y_{t+1}^j\}$, random samples $\{X_{t+1}^j | i = 1 \sim p(x_{t+1} | y_{t+1}^j)\}$ from Algorithm 1 delivers an SMC approximation of $p(dx_{t+1} | y_{t+1}^j)$ given as

$$\hat{p}(dx_{t+1} | y_{t+1}^j) = \frac{1}{N} \sum_{j=1}^{N} \delta_{X_{t+1}^j} (dx_{t+1}) \quad (53)$$

where $\hat{p}(dx_{t+1} | y_{t+1}^j)$ is an SMC approximation of $p(dx_{t+1} | y_{t+1}^j)$. Substituting (53) and (29) into (52b) yields (49c), which completes the proof.

**Result 4.20:** An SMC approximation of the PFIM for Model 4.13 is obtained by substituting (49a) through (49c) into (41) in Lemma 4.19 into (4) in Lemma 3.7 such that

$$\tilde{J}_{t+1} = \tilde{D}_t^{22} - [\tilde{D}_t^{12}]^T (\tilde{J}_t + \tilde{D}_t^{11})^{-1} \tilde{D}_t^{12}, \quad (54)$$

where $\tilde{J}_{t+1}$ is an SMC approximation of $J_{t+1}$. Applying matrix inversion lemma 5.1 in (54) gives an SMC approximation of the PCRLB, such that

$$\tilde{J}_{t+1}^{-1} = [\tilde{D}_t^{22}]^{-1} - [\tilde{D}_t^{22}]^{-1}[\tilde{D}_t^{12}]^T [\tilde{D}_t^{12}]^{-1} - [\tilde{D}_t^{12}]^{-1}[\tilde{D}_t^{22}]^{-1}, \quad (55)$$

where $\tilde{J}_{t+1}$ is an SMC approximation of $J_{t+1}$ in (2).

**V. Final Algorithm**

Algorithms 2 and 3 give the procedure for computing an SMC approximation of the PCRLB for Models 3.1 and 4.13 respectively.

**Remark 5.1:** In practice, an ensemble of $M$ measurement sequences $\{Y_{1:T} = y_{1:T}^j\}_{j=1}^M$ required by Algorithms 2 and 3 are obtained from historical process data; however, in simulations, it can be generated by simulating Models 3.1 and 4.13 $M$ times starting at i.i.d. initial states drawn from $X_0 \sim p(x_0)$. Note that this procedure also requires simulation of the true states; however, true states are not used in Algorithms 2 and 3.

For illustrative purposes, to assess the numerical reliability of Algorithms 2 and 3, a quality measure is defined as follows:

$$\Lambda_J = \frac{1}{T} \sum_{t=1}^{T} |J_t^{-1} - \tilde{J}_t^{-1}| \circ |J_t^{-1} - \tilde{J}_t^{-1}|, \quad (56)$$

where $\Lambda_J$ is the average sum of square of errors in approximating the PCRLB and $\circ$ is the Hadamard product. $\Lambda_J$ is a $n \times n$ matrix, with diagonal element $\Lambda_J(j,j)$ as the average sum of square of errors accumulated in approximating the PCRLB for state $j$, where $1 \leq j \leq n$.

---

**Algorithm 2 SMC based PCRLB for Model 3.1**

**Input:** Given Model 3.1 satisfying Assumptions 3.2 through 3.5 assume a prior pdf on $X_0$, such that $X_0 \sim p(x_0)$. Also, select algorithm parameters $T$, $N$ and $M$.

**Output:** SMC approximation of the PCRLB for Model 3.1

1: Generate and store $M$ i.i.d. sequences $\{Y_{1:T} = y_{1:T}^j\}_{j=1}^M \sim p(y_{1:T})$ of length $T$, by simulating Model 3.1 $M$ times starting at $M$ i.i.d. initial states $\{X_{0|0}^j \sim p(x_0)\}$. 
2: for $j = 1$ to $M$ do
3: for $t = 1$ to $T$ do
4: Store resampled particles $\{X_{t|1}^j = x_{t|1}^j \sim p(x_t | y_{t|1}^j)\}$ using Algorithm 1
5: Store resampled particles $\{X_{t|1}^j = x_{t|1}^j \sim p(x_{t-1} | y_{t|1}^j)\}$ using Lemma 4.12
6: end for
7: end for
8: Compute PFIM $J_0$ at $t = 0$ based on the initial target state pdf $X_0 \sim p(x_0)$. If $X_0 \sim N(x_0 | C_{x_0}, P_{0|0})$ then from Lemma 3.7 $J_0 = P_{0|0}^{-1}$. 
9: for $t = 0$ to $T - 1$ do
10: Compute an SMC estimate (30a) through (30c) in Lemma 4.14
11: Compute PCRLB $\tilde{J}_{t+1}^T$ by substituting (30a) through (30c) into (55).
12: end for

---

**VI. Convergence**

Computing the PCRLB in Lemma 3.6 involves solving the complex, multi-dimensional integrals; however, as stated earlier, for Models 3.1 and 4.13 the PCRLB cannot be solved in closed form. Algorithms 2 and 3 gives a $N$ particle and $M$ simulation based SMC approximation of the PCRLB for Models 3.1 and 4.13 respectively. It is therefore natural to question the convergence properties of the proposed numerical method. In this regard, results such as Theorem 4.5 and Remark 4.11 are important as it ensures that the proposed numerical solution does not result in accumulation of errors. It is emphasized that although Theorem 4.5 and Remark 4.11 not necessarily imply convergence of the SMC based PCRLB and MSE to its theoretical values, nevertheless, it provides a strong theoretical basis for the numerous approximations used in Algorithms 2 and 3.

From an application perspective, it is instructive to highlight that the numerical quality of the SMC based PCRLB approximation in Algorithms 2 and 3 can be made accurate by simply increasing the number of particles ($N$) and the MC simulations ($M$). The choice of $N$ and $M$ are user defined, which can be selected based on the required numerical accuracy, and available computing speed. It is important to emphasize that due to the multiple approximations involved in deriving a tractable solution, for practical purposes, with a finite $N$ and $M$, the condition $P_{t|t} - \tilde{J}_t^{-1} \gg 0$ is not guaranteed to hold for
Algorithm 3 SMC based PCRLB for Model 4.15

**Input:** Given Model 4.15 satisfying Assumptions 3.2 through 3.5 assume a prior on \( X_0 \), such that \( X_0 \sim p(x_0) \). Also, select algorithm parameters \( T, N \) and \( M \).

**Output:** SMC approximation of the PCRLB for Model 4.15

1: Generate and store \( M \) i.i.d. sequences {\( Y_{i,T} \)} \( \sum_{i=1}^{M} \approx p(y_{1:T}) \) of \( T \) by simulating Model 4.15 \( M \) times starting at \( M \) i.i.d. initial states \( \{X^1_{0:j=1}\} \). \( \sim p(x_{0}) \).
2: for \( j = 1 \) to \( T \) do
3: for \( t = 1 \) to \( M \) do
4: Store predicted particles \( \{X^i_{t-1}\} \sim p(x_{t} | y_{1:t}) \) using Algorithm 1.
5: Store resampled particles \( \{X^i_{t}\} \sim p(x_{t} | y_{1:t}) \) using Algorithm 1.
6: Store resampled particles \( \{X^i_{t-1}\} \sim p(x_{t-1} | y_{1:t-1}) \) using Lemma 4.13.
7: end for
8: end for
9: Compute PFIM \( J_0 \) at \( t = 0 \) based on the initial target state pdf \( X_0 \sim p(x_0) \). If \( X_0 \sim N(0 \mid \text{C}_{x_0} \text{F}_{0_0}) \) then from Lemma 3.7, \( J_0 = P_{0_0}^{-1} \).
10: for \( t = 0 \) to \( T - 1 \) do
11: Compute SMC estimate (49a) through (49c) in Algorithm 1.
12: Compute PCRLB \( \bar{J}_{t+1} \) by substituting (49a) through (49c) into using (55).
13: end for

all \( t \in \mathbb{N} \).

The quality of the SMC based PCRLB solution is validated next via simulation.

VII. SIMULATION EXAMPLES

In this section, two simulation examples are presented to demonstrate the utility and performance of the proposed SMC based PCRLB solution. The first example is a ballistic target tracking problem at re-entry phase. The aim of this study is three fold: first to demonstrate the performance and utility of the proposed method on a practical problem; second, to demonstrate the quality of the bound approximation for a range of target state and sensor noise variances; and third, to study the sensitivity of the involved SMC approximations to the number of particles used.

The performance of the SMC based PCRLB solution on a second example involving a uni-variate, non-stationary growth model, which is a standard non-linear, and bimodal benchmark model is then illustrated. This example is profiled to demonstrate the accuracy of the SMC based PCRLB solution for highly non-linear SSMs with non-Gaussian noise.

A. Example 1: Ballistic target tracking at re-entry

In Section IV-C an SMC based method for approximating the PCRLB was presented for non-linear SSMs with additive Gaussian state and sensor noise (See Algorithm 3). In this section, the quality of Algorithm 3 is validated on a practical problem of ballistic target tracking at re-entry phase. This particular problem has attracted a lot of attention from researchers for both theoretical and practical reasons. See [52] and the references cited therein for a detailed survey on the ballistic target tracking.

1) Model setup: Consider a target launched along a ballistic flight whose kinematics are described in a 2D Cartesian coordinate system. This particular description of the kinematics assumes that the only forces acting on the target at any given time are due to gravity and drag. All other forces such as: centrifugal acceleration, Coriolis acceleration, wind, lift force and spinning motion are assumed to have a small effect on the target trajectory. With the position and the velocity of the target at time \( t \in \mathbb{N} \) described in 2D Cartesian coordinate system as \( (X_t, H_t) \) and \( (\dot{X}_t, \dot{H}_t) \), respectively, its motion in the re-entry phase can be described by the following discrete-time non-linear SSM [13]

\[
X_{t+1} = AX_t + GF_t(X_t) + G \cdot \begin{bmatrix} 0 \\ -g \end{bmatrix} + V_t, \tag{57}
\]

where the states \( X_t \triangleq [X_t \ X_t \ H_t \ \dot{H}_t]^T \). Also, the matrices \( A \) and \( G \) are as follows

\[
A \triangleq \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ G \triangleq \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & 0 \end{bmatrix}, \tag{58}
\]

where \( \Delta T \) is the time interval between two consecutive radar measurements.

In (57) \( F_t(X_t) \) models the drag force, which acts in a direction opposite to the target velocity. In terms of the states, \( F_t(X_t) \) can be modelled as

\[
F_t(X_t) = -g \rho(H_t) \sqrt{X_t^2 + H_t^2} \begin{bmatrix} X_t \\ \dot{X}_t \end{bmatrix}, \tag{59}
\]

where \( g \) is the acceleration due to gravity; \( \beta \) is the ballistic coefficient whose value depends on the shape, mass and the cross sectional area of the target [11]; and \( \rho(H_t) \) is the density of the air, defined as an exponentially decaying function of \( H_t \), such that

\[
\rho(H_t) = \alpha_1 e^{(-\alpha_2 H_t)} \tag{60}
\]

where: \( \alpha_1 = 1.227 \text{ kg m}^{-3}, \alpha_2 = 1.09310 \times 10^{-4} \text{ m}^{-1} \) for \( H_t < 9144 \text{ m} \); and \( \alpha_1 = 1.754 \text{ kg m}^{-3}, \alpha_2 = 1.4910 \times 10^{-4} \text{ m}^{-1} \) for \( H_t \geq 9144 \text{ m} \). Note that the drag force, \( F_t(X_t) \) is the only non-linear term in the state equation. In (57) the state noise \( V_t \in \mathbb{R}^4 \) is a i.i.d. sequence of multi-variate Gaussian random vector represented as \( V_t \sim N(v_t | 0, Q_t) \), with zero mean and covariance matrix \( Q_t \) given as

\[
Q_t = \gamma I_{2 \times 2} \otimes \Theta, \quad \Theta = \begin{bmatrix} \frac{\Delta T^3}{3} & \frac{\Delta T^2}{2} \\ \frac{\Delta T^2}{2} & \Delta T \end{bmatrix}, \tag{61}
\]

where: \( \gamma \in \mathbb{R}_+; \ I_{2 \times 2} \) is a \( 2 \times 2 \) identity matrix; and \( \otimes \) is the Kronecker product. The intensity of the state noise, determined
by $\gamma$, accounts for all the forces neglected in (57), including any deviations arising due to system-model mismatch. The target measurements are collected by a conventional radar (e.g., dish radar) assumed to be stationed at the origin. The sensor readings are measured in the natural sensor coordinate system, which include range ($R_t$) and elevation ($E_t$) of the target. The radar readings $Y_t = [R_t, E_t]^T$ are related to the states $X_t$ through a non-linear observation model given below.

$$Y_t = \left[\sqrt{X_t^2 + H_t^2}, \arctan \left(\frac{H_t}{X_t}\right)\right] + W_t. \quad (62)$$

In (62) $W_t \in \mathbb{R}^2$ is an i.i.d. sequence of multi-variate Gaussian random vector represented as $W_t \sim \mathcal{N}(0, R_t)$, with zero mean and non-singular covariance matrix $R_t$ given as

$$R_t = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_e^2 \end{bmatrix}, \quad (63)$$

where $\sigma_r \in \mathbb{R}_+$ and $\sigma_e \in \mathbb{R}_+$ are the standard deviation associated with range and elevation measurements. In (62), it is assumed that the true target elevation angle lies between 0 and $\pi/2$ radians; otherwise, it suffices to add $\pi$ radians to the arctan term in (62).

**Remark 7.1:** To avoid use of a non-linear sensor model, some authors [13, 14] considered transforming the radar measurements in (62) into the Cartesian coordinate system, wherein the sensor dynamics manifest themselves into a linear model. Even though this strategy eliminates the need to handle non-linearity in sensor measurements, tracking in Cartesian coordinates couples the sensor noise across two coordinate systems and makes the noise non-Gaussian and state dependent [53]. Since the proposed method can deal with strong state and sensor non-linearities, the radar readings are monitored in natural sensor coordinates alone.

2) Simulation setup: For simulation, the model parameters are selected as given in Table I. The aim of this study is to evaluate the quality of the SMC based PCRLB solution for a range of target state and sensor noise variances. This allows full investigation of the quality of the SMC based approximation for a range of noise characteristics. The cases considered here are given in Table II. From Assumption 3.2, $\beta$ is assumed to be fixed and known a priori.

### Table I: Parameter values used in Example 1.

| Process variables | Symbol values |
|-------------------|---------------|
| accel. due to gravity | $g = 9.8 \text{ m/s}^2$ |
| ballistic coefficient | $\beta = 4000 \text{ kg.m}^{-1}.\text{s}^{-2}$ |
| radar sampling time | $\Delta T = 2 \text{ s}$ |
| total tracking time | $T = 120 \text{ s}$ |
| state noise | $V_t \sim \mathcal{N}(v_t[0], Q_t)$ |
| sensor noise | $W_t \sim \mathcal{N}(w_t[0], R_t)$ |
| noise parameters | $\gamma, \sigma_r, \sigma_e$ see Table III |
| initial states | $X_0 = \begin{bmatrix} 232 \text{ km} \\ 2.290 \cos (190^\circ) \text{ km/s} \\ 88 \text{ km} \\ 2.290 \sin (190^\circ) \text{ km/s} \end{bmatrix}$ |
| probability of detection | $Pr_d = 1$ |
| probability of false alarm | $Pr_f = 0$ |

### Table II: Cases considered for Example 1.

| Case | $\gamma$ | $\sigma_r$ | $\sigma_e$ |
|------|----------|------------|------------|
| 1    | 1.0      | 100 m      | 0.01 rad   |
| 2    | 5.0      | 100 m      | 0.017 rad  |
| 3    | 1.0      | 500 m      | 0.085 rad  |
| 4    | 5.0      | 500 m      | 0.085 rad  |

### Table III: Variable values used in Example 1.

| Process variables | Symbol values |
|-------------------|---------------|
| state noise | $V_t \sim \mathcal{N}(v_t[0], Q_t)$ |
| sensor noise | $W_t \sim \mathcal{N}(w_t[0], R_t)$ |
| noise parameters | $\gamma, \sigma_r, \sigma_e$ see Table III |
| initial states | $X_0 = \begin{bmatrix} 232 \text{ km} \\ 2.290 \cos (190^\circ) \text{ km/s} \\ 88 \text{ km} \\ 2.290 \sin (190^\circ) \text{ km/s} \end{bmatrix}$ and $P_0[1/2] = \begin{bmatrix} 0 & 20 \text{ m/s} \\ 20 \text{ m/s} & 0 \end{bmatrix}$ |
| Number of particles | $N = 1000$ |
| MC simulations | $M = 200$ |

Figure I shows a sample trajectory of the target in the $X-H$ plane along with its velocity map as a function of time, generated using Case 1 (see Table III).

3) Results: The kinematics of the ballistic target consist of nonlinear state and sensor models with additive Gaussian noise, for which the PCRLB can be approximated using Algorithm 2. First, the state and sensor models in (57) and (62), respectively, are defined as

$$f_t(X_t) = AX_t + GF_t(X_t) + G_r \begin{bmatrix} 0 \\ -g \end{bmatrix}, \quad (64a)$$

$$g_{t+1}(X_{t+1}) = \begin{bmatrix} \sqrt{X_{t+1}^2 + H_{t+1}^2} \\ \arctan \left(\frac{H_{t+1}}{X_{t+1}}\right) \end{bmatrix}. \quad (64b)$$

To compute the required gradients $\nabla X_t f_t(X_t)$ and $\nabla X_{t+1} g_{t+1}(X_{t+1})$, differentiating (57) with respect to $X_t$ and (62) with respect to $X_{t+1}$, yields

$$\nabla X_t f_t(X_t) = A + GM_t(X_t), \quad (65a)$$

$$\nabla X_{t+1} g_{t+1}(X_{t+1}) = N_t+1(X_{t+1}), \quad (65b)$$

where: $M_t(X_t)$ and $N_t+1(X_{t+1})$ in (65a) and (65b), respectively, are $2 \times 4$ matrices, whose entries are:

$$M_t(X_t)[1, 1] = 0, \quad M_t(X_t)[2, 1] = 0, \quad (66a)$$

$$M_t(X_t)[1, 2] = -\frac{g}{2\beta} \rho(H_t) \begin{bmatrix} 2X_t^2 + H_t^2 \\ \sqrt{X_t^2 + H_t^2} \end{bmatrix}, \quad (66c)$$

$$M_t(X_t)[2, 2] = -\frac{g}{2\beta} \rho(H_t) \begin{bmatrix} X_t \dot{H}_t \\ \sqrt{X_t^2 + H_t^2} \end{bmatrix}, \quad (66d)$$

$$M_t(X_t)[1, 3] = \frac{g\alpha \gamma}{2\beta} \rho(H_t) \begin{bmatrix} X_t \dot{H}_t \\ \sqrt{X_t^2 + H_t^2} \end{bmatrix} \dot{X}_t, \quad (66e)$$

$$M_t(X_t)[2, 3] = \frac{g\alpha \gamma}{2\beta} \rho(H_t) \begin{bmatrix} X_t \dot{H}_t \\ \sqrt{X_t^2 + H_t^2} \end{bmatrix} \dot{H}_t, \quad (66f)$$
Case 1: fair comparison of all the cases, the parameters required by follows the theoretical bound at all tracking time instants. Clearly, the approximate bound for both the position and velocity of the target in both X and H coordinates accurately based approximate bound against the theoretical PCRLB. Note that the high values of the PCRLB in Figure 2(a) highlights tracking difficulties as the target approaches the ground.

To evaluate the numerical quality of Algorithm 1, we compare the SMC based PCRLB solution against the theoretical values. The theoretical bound is computed using an ensemble of the true state trajectories, simulated using (57) (see [11], [13] for further details). Here we compare the square root of the diagonal elements of the theoretical PCRLB matrix $J_{t}$ and its approximation $J_{t}^{-1}$ for all $t \in [0, T]$. The results are summarized next for the cases given in Table III. For fair comparison of all the cases, the parameters required by Algorithm 3 are specified as given in Table III.

Case 1: Figure 2(a) compares the square root of the SMC based approximate bound against the theoretical PCRLB. Clearly, the approximate bound for both the position and velocity of the target in both X and H coordinates accurately follows the theoretical bound at all tracking time instants. Note that the high values of the PCRLB in Figure 2(a) highlights tracking difficulties as the target approaches the

$$M_t(X_t)[1, 4] = M_t(X_t)[2, 2],$$

$$M_t(X_t)[2, 4] = \frac{9}{2 \beta} \rho(H_t) \left[ \frac{X_t^2 + 2H_t^2}{X_t^2 + H_t^2} \right];$$

and:

$$N_{t+1}(X_{t+1})[1, 1] = \frac{X_{t+1}}{X_{t+1}^2 + H_{t+1}^2},$$

$$N_{t+1}(X_{t+1})[2, 1] = \frac{H_{t+1}}{X_{t+1}^2 + H_{t+1}^2},$$

$$N_{t+1}(X_{t+1})[1, 2] = 0,$$

$$N_{t+1}(X_{t+1})[2, 2] = 0,$$

$$N_{t+1}(X_{t+1})[1, 3] = \frac{H_{t+1}}{X_{t+1}^2 + H_{t+1}^2},$$

$$N_{t+1}(X_{t+1})[2, 3] = \frac{X_{t+1}}{X_{t+1}^2 + H_{t+1}^2},$$

$$N_{t+1}(X_{t+1})[1, 4] = 0,$$

$$N_{t+1}(X_{t+1})[2, 4] = 0.$$

TABLE IV: Average sum of square of errors in approximating the PCRLB for the states in Example 1, under the cases in Table III

| $\Lambda_J$ values | Case 1 | Case 2 | Case 3 | Case 4 |
|-------------------|--------|--------|--------|--------|
| $\Lambda_J(1, 1)$ ($10^{-16}$) | 9.30   | 50.7   | 5.87   | 130    |
| $\Lambda_J(2, 2)$ ($10^{-11}$) | 4.50   | 2.06   | 7.08   | 46.2   |
| $\Lambda_J(3, 3)$ ($10^{-5}$) | 3.56   | 23.1   | 2.96   | 100    |
| $\Lambda_J(4, 4)$ ($10^{-13}$) | 8.63   | 24.8   | 19.6   | 122    |

Fig. 1: Sample trajectory showing position and velocity of the target at re-entry phase.
Fig. 2: Results for Simulation Example 1.
Remark 7.2: Note that in [34], a similar ballistic target tracking problem at re-entry phase was considered to illustrate the use of an EKF and UKF based method in approximating the theoretical PCRLB. Unlike the non-linear sensor model considered here (see [62]), [34] used the change of coordinates method to obtain a linear sensor model representation. It is important to highlight that even with a linear sensor model, the EKF and UKF based method yields a biased estimate of the PCRLB for the target states (see Figures 4 through 7 in [34]). Whereas, under a more challenging situation, as one considered here, the SMC based method yields an unbiased estimate of the PCRLB (see Figures 2(a) through 2(d) and Table IV). This highlights the advantages of the SMC based method (both in terms of the accuracy and applicability) over the EKF and UKF based PCRLB in presence of strong system or sensor non-linearities.

Next we study the sensitivity of the involved SMC approximations to the number of particles used. In Figure 2(f) approximate PCRLB bounds are compared against the theoretical PCRLB for different values of $N$. The results are obtained by varying $N$ in Algorithm 1. From Figure 2(f) it is clear that by simply increasing $N$, which is a tuning parameter in Algorithm 1, the quality of the SMC approximations can be significantly improved. For all the simulation cases, the number of Monte Carlo simulations was selected as $M = 200$ (see Table III). Computation of a single Monte Carlo simulation took 0.69 seconds on a 3.33 GHz Intel Core i5 processor running on Windows 7. Note that the reported absolute execution time is solely for instructive purposes and is not intended to reflect on the true computational complexity of the proposed algorithm. Collectively, from Figures 2(a) through 2(f) it is evident that the SMC based method is accurate in approximating the theoretical PCRLB for a range of target state and sensor noise variances.

B. Example 2: A non-linear and non-Gaussian system

The aim of this study is to demonstrate the effectiveness of the proposed SMC based method in approximating the PCRLB in presence of a non-Gaussian noise.

1) Model setup: A more challenging situation is considered in this section that involves the following discrete-time, univariate non-stationary growth model

$$X_{t+1} = \frac{X_t}{2} + \frac{25X_t}{1 + X_t^2} + 8\cos(1.2t) + V_t,$$

$$Y_t = \frac{X_t^2}{20} + W_t,$$

where $V_t \in \mathbb{R}$ is an i.i.d. sequence following a Gaussian distribution, such that $V_t \sim N(w_t|0,R_t)$, while for Case 2, $W_t \in \mathbb{R}$ is again an i.i.d sequence, but follows a Rayleigh distribution, such that $W_t \sim \mathcal{R}(w_t|\lambda_t)$. For both the cases, the sensor noise variance $R_t = 1 \times 10^{-3}$ $\forall t \in [1,T]$ is considered. Here Case 2 represents a much more challenging situation, where estimation is considered under a non-Gaussian sensor noise. For fair comparison, $M = 200$ and $N = 100$ are selected.

3) Results: Case 1: Comparison of the approximate and the theoretical PCRLB for the Gaussian sensor noise case is given in Figure 3. The results suggest that for the chosen $N$, the approximate PCRLB almost exactly follows the theoretical PCRLB at all filtering time instants. The same is reflected in the error value computed using (50), which is $\Lambda_J = 4.19 \times 10^{-9}$.

Case 2: Figure 3 compares the approximate PCRLB solution against the theoretical PCRLB for the Rayleigh sensor noise case. Although the approximation almost exactly follows the theoretical solution, compared to Case 1, the approximation is relatively coarser at certain time instants. This highlights the issues associated with estimation under non-Gaussian noise with limited $N$. Finally, the $\Lambda_J$ value for Case 2 is $4.62 \times 10^{-8}$, which is within an order of the value reported for Case 1.

The simulation study clearly illustrates the efficacy of the proposed method in approximating the PCRLB for non-linear SSMs with non-Gaussian noise.

VIII. DISCUSSIONS

The simulation results in Section VII demonstrate the utility and performance of the SMC based PCRLB approximation method developed in this paper. It is important to highlight that despite of the many convergence results discussed in Section VI, the choice of an SMC method plays a crucial role in determining the quality of the PCRLB approximation. Here, the use of a sequential-importance-resampling (SIR) filter of [41], [45] is motivated by the fact that it is relatively less sensitive to large state noise and is computationally less expensive. Furthermore, the importance weights are easily evaluated and the importance functions can be easily sampled [11]; however, other algorithms such as Auxiliary-SIR (ASIR) [42] or Regularized PF (RPF) [54] algorithm can also be
used in place of SIR, as long as they are consistent with the approach developed herein.

An appropriate choice of the resampling method in Algorithm 1 is also crucial as it can substantially improve the quality of the approximations. The choice of the systematic resampling is supported by an easy implementation procedure and the low-order of computational complexity $O(N)$ [7]. Other resampling schemes such as stratified sampling [55] and residual sampling [56] can also be used as an alternative to systematic resampling in the proposed framework.

In summary, with the aforementioned options, coupled with the user-defined choice of the parameters $N$ and $M$, an SMC based PCRLB approximation approach provides an efficient control over the numerical quality of the solution.

IX. Conclusions

In this paper a numerical method to recursively approximate the PCRLB in (1) for a general discrete-time, non-linear SSMs operating with $P_{t|t-1} = 1$ and $P_{t|T} = 0$ is presented. The presented method is effective in approximating the PCRLB, when the true states are hidden or unavailable. This has practical relevance in situations; wherein, the test-data consist of only sensor readings. The proposed approach makes use of the sensor readings to estimate the hidden true states, using an SMC method. The method is general and can be used to compute the lower bound for non-linear dynamical systems, with non-Gaussian state and sensor noise. The quality and utility of the SMC based PCRLB approximation was validated on two simulation examples, including a practical problem of ballistic target tracking at re-entry phase. The analysis of the numerical quality of the SMC based PCRLB approximation was investigated for a range of target state and sensor noise variances, and with different number of particles. The proposed method exhibited acceptable and consistent performance in all the simulations. Increasing the number of particles was in particular, found to be effective in reducing the errors in the PCRLB estimates. Finally, some of the strategies for improving the quality of the SMC based approximations were also discussed.

The current paper assumes the model parameters to be known a priori; however, for certain applications, this assumption might be a little restrictive. Future work will focus on extending the results of this work to handle such situations. Furthermore, use of SMC method in approximating the modified versions of the PCRLB, which allow tracking in situations, such as: target generated measurements; measurement origin uncertainty; cluttered environments; and Markovian models will also be considered.

Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council (NSERC), Canada.

References

[1] P. Tichavský, C. Muravchik, and A. Nehorai, “Posterior Cramér-Rao bounds for discrete-time non-linear filtering,” IEEE Transactions on Signal Processing, vol. 46, no. 5, pp. 1386–1396, 1998.

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

[3] N. Gordon, D. Salmond, and C. Ewing, “Bayesian state estimation for tracking and guidance using the bootstrap filter,” Journal of Guidance, Control and Dynamics, vol. 18, no. 6, pp. 1434–1443, 1995.

[4] C. Chang and J. Tabaczyński, “Application of state estimation to target tracking,” IEEE Transactions on Automatic Control, vol. 29, no. 2, pp. 98–109, 1984.

[5] R. Dearden, T. Willeke, R. Simmons, V. Verma, F. Hutter, and S. Thrun, “Real-time fault detection and situational awareness for rovers: report on the Mars technology program task,” in Proceedings of the IEEE Aerospace Conference, Montana, USA, 2004, pp. 826–840.

[6] N. de Freitas, R. Dearden, F. Hutter, R. Menendez, J. Mutch, and D. Poole, “Diagnosis by a waiter and a Mars explorer,” Proceedings of the IEEE, vol. 92, no. 3, pp. 455–468, 2004.

[7] A. Doucet, N. de Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice. Springer–Verlag, New York, 2001, ch. On sequential Monte Carlo methods.

[8] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002.

[9] N. Simandl, J. Krállové, and P. Tichavský, “Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time non-linear dynamic systems,” Automatica, vol. 37, no. 11, pp. 1703–1716, 2001.

[10] H. Trees, Detection, Estimation and Modulation Theory–Part I. Wiley, New York, 1968, ch. Classical detection and estimation theory.

[11] B. Ristic, M. Arulampalam, and N. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, Boston, 2004, ch. Suboptimal non-linear filters.

[12] N. Bergman, Sequential Monte Carlo Methods in Practice. Springer–Verlag, New York, 2001, ch. Posterior Cramér-Rao bounds for sequential estimation.

[13] A. Farina, B. Ristic, and D. Benvenuti, “Tracking a ballistic target: comparison of several non-linear filters,” IEEE Transactions on Aerospace and Electronic Systems, vol. 38, no. 3, pp. 854–867, 2002.

[14] N. Bergman, L. Ljung, and F. Gustafsson, “Terrain navigation using Bayesian statistics,” IEEE Control Systems Magazine, vol. 19, no. 3, pp. 33–40, 1999.

[15] A. Nehorai and M. Hawkes, “Performance bounds for estimating vector systems,” IEEE Transactions on Signal Processing, vol. 48, no. 6, pp. 1737–1749, 2000.

[16] J. Glass and L. Smith, “MIMO radar resource allocation using posterior Cramér-Rao lower bounds,” in Proceedings of the IEEE Aerospace Conference, Montana, USA, 2011, pp. 1–9.

[17] M. Hernandez, T. Kirubarajan, and Y. Shalom, “Multisensor resource deployment using posterior Cramér-Rao bounds,” IEEE Transactions on Aerospace and Electronic Systems, vol. 40, no. 2, pp. 399–416, 2004.

[18] F. Farshidi, S. Sirouspour, and T. Kirubarajan, “Optimal positioning of multiple cameras for object recognition using Cramér-Rao lower bound,” in Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, USA, 2006, pp. 934–939.

[19] J. Passerieux and D. V. Cappel, “Optimal observer maneuver for bearings-only tracking,” IEEE Transactions on Aerospace and Electronic Systems, vol. 34, no. 3, pp. 777–788, 1998.

[20] J. Hellfert and D. Mudgett, “Optimal observer trajectories for bearings only tracking by minimizing the trace of the Cramér-Rao lower bound,” in Proceedings of the IEEE Conference on Decision and Control, San Antonio, USA, 1993, pp. 936–939.

[21] Y. Chou, B. Bobrovsky, and M. Zakai, “A lower bound on the estimation error for Markov processes,” IEEE Transactions on Automatic Control, vol. 20, no. 62, pp. 785–784, 1975.

[22] J. Galdos, “A Cramér-Rao bound for multi-dimensional discrete-time dynamical systems,” IEEE Transactions on Automatic Control, vol. 25, no. 1, pp. 117–119, 1980.

[23] T. Doerschuk, “A Cramér-Rao bound for discrete-time non-linear filtering problems,” IEEE Transactions on Automatic Control, vol. 40, no. 8, pp. 1465–1469, 1995.

[24] T. Kerr, “Status of CR like bounds for non-linear filtering,” IEEE Transactions on Aerospace and Electronic Systems, vol. 25, pp. 590–601, 1989.

[25] M. Hernandez, A. Mars, N. Gordon, S. Maskell, and C. Reed, “Cramér-Rao bounds for non-linear filtering with measurement origin uncertainty,” in Proceedings of the 5th International Conference on Information Fusion, Maryland, USA, 2002, pp. 18–15.
