Applied Soft Computing 72 (2018) 647–665

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Conditional predictive Bayesian Cramér-Rao Lower Bounds for prognostic algorithms design

David E. Acuña [a,*], Marcos E. Orchard [a], Raimundo J. Saona [b]

[a] Department of Electrical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, Av. Tupper 2007, Santiago, Chile
[b] Department of Mathematical Engineering, Faculty of Mathematical and Physical Sciences, University of Chile, Av. Beauchef 851, North Tower, Floor 4, Santiago, Chile

ARTICLE INFO

Article history:
Received 30 September 2017
Received in revised form 18 January 2018
Accepted 24 January 2018
Available online 10 February 2018

Keywords:
Prognostics and health management
Prognostic algorithm design
Bayesian Cramér-Rao Lower Bounds
Particle filters
Battery end-of- discharge

ABSTRACT

System states are related, directly or indirectly, to health condition indicators. Indeed, critical system failures can be efficiently characterized through a state space manifold. This fact has encouraged the development of a series of failure prognostic frameworks based on Bayesian processors (e.g. particle or unscented Kalman filters), which efficiently help to estimate the Time-of-Failure (ToF) probability distribution in nonlinear, non-Gaussian, systems with uncertain future operating profiles. However, it is still unclear how to determine the efficacy of these methods, since the Prognostics and Health Management (PHM) community has not developed rigorous theoretical frameworks that could help to define proper performance indicators. In this regard, this article introduces novel prognostic performance metric based on the concept of Bayesian Cramér-Rao Lower Bounds (BCRLBs) for the predicted state mean square error (MSE), which is conditional to measurement data and model dynamics; providing a formal mathematical definition of the prognostic problem. Furthermore, we propose a novel step-by-step design methodology to tune prognostic algorithm hyper-parameters, which allows to guarantee that obtained results do not violate fundamental precision bounds. As an illustrative example, both the predictive BCRLB concept and the proposed design methodology are applied to the problem of End-of-Discharge (EoD) time prognostics in lithium-ion batteries.

1. Introduction

Failure prognostic algorithms use long-term predictions to describe the evolution in time of a condition indicator with the purpose of estimating the Time-of-Failure (ToF) of a system, or the Remaining Useful Life (RUL) of a failing component. These long-term predictions are made using a thorough understanding of the underlying degradation processes and the anticipated future usage, as well as an effective characterization of all associated uncertainty sources.

Probability-based methods are preferred when characterizing uncertainty sources in failure prognostic algorithms [1], since they provide a well-known mathematical framework that can be directly applied to data analysis, modeling identification, and parameter estimation problems. Particularly, Bayesian approaches [2] stand out as a suitable option for online characterization of uncertainty sources in failure prognostic algorithms, since they provide a sound theoretical framework for the implementation of filtering, smoothing, and prediction methods in nonlinear, non-Gaussian, dynamic processes [3]. For this reason, many Bayesian state estimation methods have been applied in the past to characterize initial conditions for the long-term prediction problem [4], while others were used to characterize future loading (or stress) profiles [5].

In failure prognostics, an effective characterization of future uncertainty sources is important because we require to avoid catastrophic events and take preventive measures [6,4]. Although this problem can be solved, if we assume that both the actual system condition and degradation model are known, by performing Monte Carlo (MC) simulation [7], the computational cost associated with this method is humongous and nearly impossible to handle for real-time decision-making processes (such as the ones found in condition-based maintenance schemes). The Prognostic and Health Management (PHM) community has chosen particle-filtering-based algorithms [8] as the *de*

* Corresponding author.
E-mail address: davacuna@ing.uchile.cl (D.E. Acuña).

https://doi.org/10.1016/j.asoc.2018.01.033
1568-4946/

facto alternative to MC, since particle-filters (PFs) offer an interesting balance between efficiency and efficacy in state estimation problems. However, it is still not clear how to measure the efficacy of particle-filtering-based prognostic methods in terms of the generated results, because the PHM community has not yet established adequate performance metrics. Indeed, prognostic algorithms lack standard definitions and suffer from ambiguous and inconsistent interpretations [9]. This lack of standards is in part due to varied end-user requirements, a significant fact when considering that forecasting is a topic of interest for a number of different domains, including aerospace, automotive, electronics, finance, medicine, nuclear power, and weather. The definition of standardized performance metrics (and not simply performance *indicators*) is still a matter of debate.

The general agreement is that better algorithms will exhibit better accuracy (related to estimates biases) and precision (related to estimates variances). This idea sounds natural and intuitive. However, it is easy to artificially "improve" the precision exhibited by an algorithm by modifying hyper-parameters of the model that defines the evolution of the state over time (state transition model). It is only natural to wonder which is the fundamental limit for these "improvements". May I be generating precise, although biased, estimates of the time-of-failure? The latter could be catastrophic in terms of decision-making processes related to maintenance scheduling.

This research effort aims at introducing a novel prognostic performance metric based on the concept of Bayesian Cramér-Rao Lower Bounds (BCRLBs) for the predicted state mean square error (MSE), which is conditional to measurement data and model dynamics; providing a formal mathematical definition of the prognostic problem. Furthermore, we propose a novel step-by-step design methodology to tune prognostic algorithm hyper-parameters, which allows to guarantee that obtained results do not violate fundamental precision bounds. In this regard, the contributions of this article are: (i) definition of a more general notion of algorithm efficacy in prognostics, based on theoretical bounds that characterize the quality of long-term predictions, and (ii) the design of a step-by-step methodology aimed at tuning the parameters of prognostic algorithms; guaranteeing that the precision of obtained results does not violate these fundamental bounds. As an illustrative example, both the predictive BCRLB concept and the proposed design methodology are applied to the problem of End-of-Discharge (EoD) time prognostics in lithium-ion batteries.

The article structure is as follows. BCRLBs are introduced in Section 2. Section 3 presents a novel prognostic performance metric based on BCRLBs and a step-by-step methodology for prognostic algorithm design. Section 4 shows the application of the proposed metrics and design methodology to the EoD problem in lithium-ion batteries and Section 5 presents main conclusions.

2. Cramér-Rao Lower Bounds

The Cramér-Rao Lower Bound (CRLB) [10,11] is a fundamental limit that establishes a lower bound for the mean square error (MSE) of any estimator. The most conventional version of this bound was developed for the assessment of the performance of unbiased estimators for unknown deterministic parameters. Later, Van Trees developed an analogous bound applicable to the case of random parameters, the so called Bayesian Cramér-Rao Lower Bound (BCRLB) [12]. The BCBR does not require the assumption of unbiased estimators.

2.1. Bayesian Cramér-Rao Lower Bounds

Let $x \in \mathbb{R}^{n_x}$ a vector of random parameters to be estimated and $y \in \mathbb{R}^{n_y}$ a random vector of observations. Let $\hat{x}(y)$ be an estimator of x obtained as a function of the observations y. The Bayesian Cramér-Rao inequality [12] establishes that

$$\mathbb{E}_{p(x,y)}\{[\hat{x}(y)-x][\hat{x}(y)-x]^T\} \geq J^{-1} \tag{1}$$

where $p(x, y)$ is a joint probability density function and J is the Bayesian Information Matrix (BIM) (called Fisher Information Matrix in the conventional setting of deterministic parameter estimation), defined as

$$J = \mathbb{E}_{p(x,y)}\{-\Delta_x^x \log p(x,y)\} \tag{2}$$

The operator Δ denotes the second-order derivative

$$\Delta_x^y = \nabla_x \nabla_y{}^T, \tag{3}$$

and with ∇ denoting the gradient operator.

2.2. BCRLBs for Discrete-Time Dynamical Systems

Bayesian approaches [2] constitute a suitable option for online characterization of uncertainty sources and degradation processes that affect the condition of nonlinear dynamic processes, via a state-space representation. In this regard, let us consider $\{X_k, k \in \mathbb{N}\}$ a first order Markov process denoting a n_x- dimensional system state vector with initial distribution $p(x_0)$ and transition probability $p(x_k|x_{k-1})$. Also, let $\{Y_k, k \in \mathbb{N}\backslash\{0\}\}$ denote n_y-dimensional conditionally independent noisy observations. Then,

$$x_k = f(x_{k-1}, \omega_{k-1}) \tag{4}$$

$$y_k = g(x_k, v_k), \tag{5}$$

where ω_k and v_k denote independent, not necessarily Gaussian, random vectors.

According to [13], there are different versions of the BCRLB for discrete-time dynamical system that may be used as a lower bound for the MSE. Let $x_{0:k} = \begin{bmatrix} x_0{}^T & x_1{}^T & \dots & x_k{}^T \end{bmatrix}^T$ and $y_{1:k} = \begin{bmatrix} y_1{}^T & y_2{}^T & \dots & y_k{}^T \end{bmatrix}^T$ denote a collection of augmented states and measurement vectors up to time k. The estimator of x_k is denoted as $\hat{x}_k(y_{1:k})$, which is a function of the measurement sequence $y_{1:k}$. Let also denote $\hat{x}_{0:k}(y_{1:k})$ the estimator of the whole state trajectory $x_{0:k}$. The inequalities regarding these versions of BCRLB's are summarized below.

A. *Joint unconditional BCRLB*

$$\mathbb{E}_{p(x_{0:k},y_{1:k})}\{[\hat{x}_{0:k}(y_{1:k})-x_{0:k}][\hat{x}_{0:k}(y_{1:k})-x_{0:k}]^T\} \geq J_{0:k}^{-1} \tag{6}$$

$$J_{0:k}^{-1} = \mathbb{E}_{p(x_{0:k},y_{1:k})}\{-\Delta_{x_{0:k}}^{x_{0:k}} \log p(x_{0:k}, y_{1:k})\} \tag{7}$$

B. *Marginal unconditional BCRLB*

$$\mathbb{E}_{p(x_k,y_{1:k})}\{[\hat{x}_k(y_{1:k}) - x_k][\hat{x}_k(y_{1:k}) - x_k]^T\} \geq J_k^{-1} \tag{8}$$

$$J_k^{-1} = \mathbb{E}_{p(x_k,y_{1:k})}\{-\Delta_{x_k}^{x_k} \log p(x_k, y_{1:k})\} \tag{9}$$

C. *Joint conditional BCRLB*

$$\mathbb{E}_{p(x_{0:k},y_k|y_{1:k-1})}\{[\hat{x}_{0:k}(y_{1:k}) - x_{0:k}][\hat{x}_{0:k}(y_{1:k}) - x_{0:k}]^T\} \geq J_{0:k}(y_{1:k-1})^{-1} \tag{10}$$

$$J_{0:k}(y_{1:k-1})^{-1} = \mathbb{E}_{p(x_{0:k},y_k|y_{1:k-1})}\{-\Delta_{x_{0:k}}^{x_{0:k}} \log p(x_{0:k}, y_k|y_{1:k-1})\} \tag{11}$$

D. *Marginal conditional BCRLB*

$$\mathbb{E}_{p(x_k,y_k|y_{1:k-1})}\{[\hat{x}_k(y_{1:k}) - x_k][\hat{x}_k(y_{1:k}) - x_k]^T\} \geq J_k(y_{1:k-1})^{-1} \tag{12}$$

$$J_k(y_{1:k-1})^{-1} = \mathbb{E}_{p(x_k,y_k|y_{1:k-1})}\{-\Delta_{x_k}^{x_k} \log p(x_k, y_k|y_{1:k-1})\} \tag{13}$$

The aforementioned bounds can be classified according to two main criteria. On the one hand, the bound is said to be *joint* if it restricts the MSE of the whole state trajectory $x_{0:k}$, whereas if it solely limits the MSE of the state vector x_k, the bound is said to be *marginal*. On the other hand, if the bound considers measurements $y_{1:k-1}$ as a random vector, it is said to be *unconditional*, whereas if $y_{1:k-1}$ is a vector of known measurements, it is said to be *conditional*.

[14] proposed an elegant way for computing the marginal unconditional BCRLB J_k^{-1} (see Eq. (9)) without manipulating large matrices at each time instant k in the following manner:

$$J_{k+1} = D_k^{22} - D_k^{21}(J_k + D_k^{11})^{-1} D_k^{12} \tag{14}$$

where

$$D_k^{11} = \mathbb{E}\{-\Delta_{x_k}^{x_k} \log p(x_{k+1}|x_k)\} \tag{15}$$

$$D_k^{12} = \mathbb{E}\{-\Delta_{x_k}^{x_{k+1}} \log p(x_{k+1}|x_k)\} = (D_k^{21})^T \tag{16}$$

$$D_k^{22} = \mathbb{E}\{-\Delta_{x_{k+1}}^{x_{k+1}}[\log p(x_{k+1}|x_k) + \log p(y_{k+1}|x_{k+1})]\} \tag{17}$$

$$= D_k^{22,a} + D_k^{22,b}. \tag{18}$$

with expectations taken with respect to $p\,(x_{0:k+1}, y_{1:k+1})$. It is important to remark that the marginal unconditional BCRLB considers random measurement vectors. [15] introduced the marginal conditional BCRLB $J_k\,(y_{1:k-1})^{-1}$ (see Eq. (13)) and also developed en elegant recursive way for its computation, in the following manner:

$$J_{k+1}(y_{1:k}) = B_k^{22} - B_k^{21}(J_k^A(Y_k) + B_k^{11})^{-1} B_k^{12}, \tag{19}$$

where

$$B_k^{11} = \mathbb{E}\{-\Delta_{x_k}^{x_k} \log p(x_{k+1}|x_k)\} \tag{20}$$

$$B_k^{12} = \mathbb{E}\{-\Delta_{x_k}^{x_{k+1}} \log p(x_{k+1}|x_k)\} = (B_k^{21})^T \tag{21}$$

$$B_k^{22} = \mathbb{E}\{-\Delta_{x_{k+1}}^{x_{k+1}}[\log p(x_{k+1}|x_k) + \log p(y_{k+1}|x_{k+1})]\} \tag{22}$$

$$= B_k^{22,a} + B_k^{22,b}, \tag{23}$$

with expectations taken with respect to $p\,(x_{0:k+1}, y_{k+1}|\, y_{1:k})$. On the other hand, $J_k^A(y_{1:k})$ is defined as the auxiliary BIM matrix for x_k, being its inverse equal to the $n_x \times n_x$ lower-right block of the inverse of the auxiliary BIM matrix $I_k^A(y_{1:k})$, where

$$I_k^A(y_{1:k}) = \mathbb{E}_{p(x_{0:k}|y_{1:k})}\{-\Delta_{x_{0:k}}^{x_{0:k}} \log p(x_{0:k}|y_{1:k})\}. \tag{24}$$

3. Methodology for prognostic algorithm design

Let us assume that we require to implement a probability-based prognostic algorithm to measure the risk of future usage for failing equipment in real-time. Let $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ be a vector of hyper-parameters that allows to configure any implementation of this probability-based prognostic algorithm. We aim at defining a step-by-step methodology that could help to tune these hyper-parameters to maximize the efficacy of the prognostic algorithm, subject to efficiency constrains (typically imposed by hardware specifications, maximum processing time, and/or computational cost). It is important to note that modifications on certain hyper-parameters will have a positive impact on the efficacy of the algorithm, while in other cases most of the impact can be measured in terms of an improvement on the efficiency. In this regard, we will group the components of the vector θ in two clusters of hyper-parameters: those that primarily affect the efficiency of the algorithm (conveniently arranged in the vector $\theta_A \in \Theta_A \subseteq \mathbb{R}^{n_{\theta_A}}$, $n_{\theta_A} < n_\theta$), and those that primarily have impact on the quality of obtained results (arranged in the vector $\theta_B \in \Theta_B \subseteq \mathbb{R}^{n_{\theta_B}}$, where $\theta^T = [\theta_A^T \quad \theta_B^T]$ and $n_{\theta_A} + n_{\theta_b} = n_\theta$). While θ_A is typically tuned to meet efficiency constraints, it is still unclear how to choose adequate values for the components of θ_B.

In this regard, we hereby propose an *ad hoc* metric, based on the Bayesian Cramér-Rao Lower Bound concept, which allows to measure the performance of the failure prognostic algorithm conditional to a realization of θ. Furthermore, we present a design methodology for

prognostic algorithms that uses predictive BCRLBs to determine fundamental limits of predicted state MSEs (at any future time instant) and a feasible region $\bar{\Theta}_B \subset \Theta_B$ of values for the hyper-parameter vector θ_B, assuming that θ_A is chosen to meet efficiency constraints. This feasible region is thereby characterized by all values of θ_B for which the predictive state MSE does not violate the corresponding predictive BCRLB (at any future time instant). A first attempt towards the incorporation of these concepts can be found in [16], where the problem of designing a prognostic algorithm to compute the Remaining Useful Life (RUL) of lithium-ion batteries (as a function of the State-of-Health) considered Bayesian bounds that were unconditional to the acquired measurements. We now extend this procedure to the case where the precision of long-term state predictions is a function of a set of known observations.

The methodology for prognostic algorithm design can be summarized as follows:

(1) Choose θ_A such that efficiency specifications are met. Compute (recursively) Bayesian Cramér-Rao bounds for the predicted state MSE (also referred to as predictive BCRLBs), starting from time k_p and up to a time prediction horizon defined by $k_h > k_p$.
(2) Choose realizations of the hyper-parameter vector $\theta_B \in \Theta_B$. You may use sampling schemes to obtain these realizations from a prior distribution.
(3) Execute the prognostic algorithm, conditional to each one of the obtained realizations for θ. Discard realizations that generate predicted state MSEs smaller than the predictive BCRLB at any time $k_p < k < k_h$.
(4) For all realizations of θ that were not discarded in Step 3, use the associated implementation of the prognostic algorithm to compute the evolution of the predicted state MSE over time. Compute a weighted average of the ℓ^1 distances between the MSE curves (per component of the state vector) and the corresponding BCRLB curves over time. Choose $[\theta_A^T \;\; \hat{\theta}_B^T]$ as the realization that minimizes the aforementioned weighted average. Compute the predicted ToF probability mass function (PMF) conditional to $[\theta_A^T \;\; \hat{\theta}_B^T]$.
(5) Explore the impact associated with a relaxation in soft efficiency constraints: Modify θ_A to allow less efficient algorithm implementations. Go through Steps 1–4, and assess the impact on the resulting ToF PMF using a metric of choice. Iterate until either the impact on the resulting ToF PMF is negligible or hard efficiency limitations are met.

This step-by-step design methodology for prognostic algorithms requires a formal definition for Bayesian Cramér-Rao Lower Bounds for the predicted state MSE, as well as a feasible procedure to compute this bound recursively. This formal definition is now presented, as part of the contributions of this research effort. Proofs for Theorem 3.1 and Theorem 3.2 can be found in Appendices A and B, respectively. Proofs for both theorems were inspired on concepts described in [15,17] . For completeness purposes, we also include the formal definition of the Time-of-Failure PMF.

3.1. Conditional predictive Bayesian Cramér-Rao Lower Bounds

Among the different expressions for Bayesian Cramér-Rao bounds currently available in the literature, it is important to remark that [17] presents an unconditional predictive version that may be applicable to failure prognostic algorithms. This unconditional BCRLB assumes an observational setting where measurements are treated as random vectors. However, in actual prognostic algorithm implementations, measurements are always assumed to be available, because it is inadequate to prognosticate a failure even before the fault could be diagnosed. In this regard, a novel performance metric based on Bayesian Cramér-Rao lower bounds for the predicted state mean square error (MSE), conditional to measurement data and model dynamics, is now introduced.

Let $x_{k_p:k} = \left[{x_{k_p}}^T \;\; {x_{k_p+1}}^T \;\; \dots \;\; {x_k}^T \right]^T$ and also x^i, $i = 1, 2, \dots, (k - k_p + 1)\, n_x$, be the ith component of the vector $x_{k_p:k}$. We will first show how to find a lower bound of the MSE associated to any estimator of $x_{k_p:k}$ (bound for the predictive state MSE) and, afterwards, we will generate a recursion that could be used to compute the bound with ease.

Let $\hat{x}_{k_p:k}(y_{1:k_p})$ be an estimator of $x_{k_p:k}$ conditional to the set of measurements acquired until the prognostic time k_p, $k > k_p$. Besides, let $\tilde{x}_{k_p:k} \triangleq \hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k}$ be the estimation error and $p_k^{cp} \triangleq p(x_{k_p:k}|y_{1:k_p})$. We will denote the second order derivative as

$$\Delta_x^y = \nabla_x \nabla_y{}^T, \tag{25}$$

where $\nabla_x = [\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_{n_x}}]$ is a gradient operator of dimensions $1 \times n_x$.

Definition 3.1. The *Conditional Predictive Bayesian Information Matrix (CPBIM)* is defined as

$$I_{cp}(x_{k_p:k}|y_{1:k_p}) \triangleq \mathbb{E}_{p_k^{cp}}\{[\nabla_{x_{k_p:k}}{}^T \log p_k^{cp}][\nabla_{x_{k_p:k}} \log p_k^{cp}]\} \tag{26}$$

The following two theorems introduce the joint and marginal versions of the CP-BCRLB, respectively. The joint version represents a bound for the predictive state MSE associated with the whole state trajectory $x_{k_p:k}$, and requires to incur in a series of expensive matrix computations that grow exponentially as $k > k_p$ increases. In contrast, the marginal version allows to obtain a bound for the predicted state MSE associated with x_k, $k > k_p$, which can be easily computed using a recursive expression.

Theorem 3.1 (Joint Conditional Predictive BCRLB). *Let us assume the following conditions about the density* p_k^{cp}:

1. p_k^{cp} *is absolutely continuous and* $\frac{\partial p_k^{cp}}{\partial x^i}$ *is absolutely integrable with respect to* $x_{k_p:k}$, *this is*

$$\int \left| \frac{\partial p_k^{cp}}{\partial x^i} \right| dx_{k_p:k} < +\infty \tag{27}$$

2. *For each* x^i, *with* $i = 1, 2, \dots, (k - k_p + 1)\, n_x$,

$$\lim_{x^i \to +\infty} x^i p(x_{k_p:k}) = \lim_{x^i \to -\infty} x^i p(x_{k_p:k}) = 0 \tag{28}$$

The MSE associated to any estimator $\hat{x}_{k_p:k}(y_{1:k_p})$ of the state trajectory $x_{k_p:k}$ is lower bounded

$$\mathbb{E}_{p_k^{cp}}\{\tilde{x}_{k_p:k}\tilde{x}_{k_p:k}^T|y_{1:k_p}\} \geq I_{cp}^{-1}(x_{k_p:k}|y_{1:k_p}), \tag{29}$$

where $I_{cp}^{-1}(x_{k_p:k}|y_{1:k_p})$ is referred to as the Joint Conditional Predictive BCRLB (JCP-BCRLB).

As stated before, we have developed a recursive formula that allows to compute the marginal version of the bound CP-BCRLB, which is now introduced.

Theorem 3.2 (Marginal Conditional Predictive BCRLB). *Let us define*

$$S_{i+1}^{i} = \mathbb{E}\{-\Delta_{x_i}^{x_i}\log p(x_{i+1}|x_i)\} \tag{30}$$

$$S_{i+1}^{i,i+1} = \mathbb{E}\{-\Delta_{x_i}^{x_{i+1}}\log p(x_{i+1}|x_i)\} \tag{31}$$

$$S_{i+1}^{i+1} = \mathbb{E}\{-\Delta_{x_{i+1}}^{x_{i+1}}\log p(x_{i+1}|x_i)\} \tag{32}$$

with $S_{i+1}^{i+1,i} = S_{i+1}^{i,i+i^T}$, $i = k_p, k_p+1, \ldots, k$. The MSE associated to x_k, is lower bounded as

$$\mathbb{E}_{p_k^{cp}}\{\tilde{x}_k\tilde{x}_k^T|y_{1:k_p}\} \geq C_k^{22} \tag{33}$$

where C_k^{22} is named as Marginal Conditional Predictive BCRLB (MCP-BCRLB), and can be recursively computed as

$$[C_k^{22}]^{-1} = S_k^k - S_k^{k,k-1}[[C_{k-1}^{22}]^{-1} + S_k^{k-1}]^{-1}S_k^{k-1,k} \tag{34}$$

considering the initial condition

$$[C_{k_p}^{22}]^{-1} = S_{k_p}^{k_p} = \mathbb{E}\{-\Delta_{x_{k_p}}^{x_{k_p}}\log p(x_{k_p}|y_{1:k_p})\}$$

3.2. Analytic computation of MCP-BCRLBs

The computation of MCP-BCRLBs requires the computation of expectations over the predictive state probability density. This fact implies that, in the case of nonlinear systems, the designer may need to perform Monte Carlo simulations to tune the hyper-parameters of a given prognostic algorithm. Although in those situations the concept of MCP-BCRLBs still helps to establish a feasible region for hyper-parameters, its implementation may require significant computational efforts.

Fortunately, MCP-BCRLBs can be analytically calculated in systems where the state transition equation has additive noise and is linear with respect to the state vector; i.e.,

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + \omega_k \\ &= A_k(u_k)\cdot x_k + B_k(u_k) + \omega_k, \end{aligned}$$

where u_k is the system's input, $A_k(u_k)$ is an n_x-dimensional square matrix, $B_k(u_k)$ is an $n_x \times 1$ matrix, and ω_k is an n_x- dimensional zero mean Gaussian random vector. Indeed, consider the case where ω_k has covariance matrix Σ_k:

$$-\log p(x_{i+1}|x_i) = c + \frac{1}{2}[x_{i+1} - f(x_i, u_i)]^T\Sigma_i^{-1}[x_{i+1} - f(x_i, u_i)]$$

$$\begin{aligned} -\nabla_{x_i}\nabla_{x_i}{}^T\log p(x_{i+1}|x_i) &= A_i(u_i)^T\Sigma_i^{-1^T}\nabla_{x_i}f(x_i, u_i) \\ &= A_i(u_i)^T\Sigma_i^{-1}A_i(u_i) \end{aligned}$$

$$\begin{aligned} -\nabla_{x_{i+1}}\nabla_{x_i}{}^T\log p(x_{i+1}|x_i) &= -A_i(u_i)^T\Sigma_i^{-1^T}\nabla_{x_{i+1}}x_{i+1} \\ &= -A_i(u_i)^T\Sigma_i^{-1} \end{aligned}$$

$$\begin{aligned} -\nabla_{x_{i+1}}\nabla_{x_{i+1}}{}^T\log p(x_{i+1}|x_i) &= \Sigma_i^{-1^T}\nabla_{x_{i+1}}x_{i+1} \\ &= \Sigma_i^{-1} \end{aligned}$$

Note that $\Sigma_i^{-1^T} = \Sigma_i^{-1}$ since the covariance matrix is assumed to be symmetric and $\nabla_y\nabla_x{}^T = \Delta_x^y$. Therefore, from Eqs. (30)–(32), the recursion for MCP-BCRLBs has analytic expressions:

$$\begin{aligned} S_{i+1}^{i} &= \mathbb{E}\{-\Delta_{x_i}^{x_i}\log p(x_{i+1}|x_i)\} = A_i(u_i)^T\Sigma_i^{-1}A_i(u_i) \\ S_{i+1}^{i,i+1} &= \mathbb{E}\{-\Delta_{x_i}^{x_{i+1}}\log p(x_{i+1}|x_i)\} = -A_i(u_i)^T\Sigma_i^{-1} \\ S_{i+1}^{i+1} &= \mathbb{E}\{-\Delta_{x_{i+1}}^{x_{i+1}}\log p(x_{i+1}|x_i)\} = \Sigma_i^{-1} \end{aligned}$$

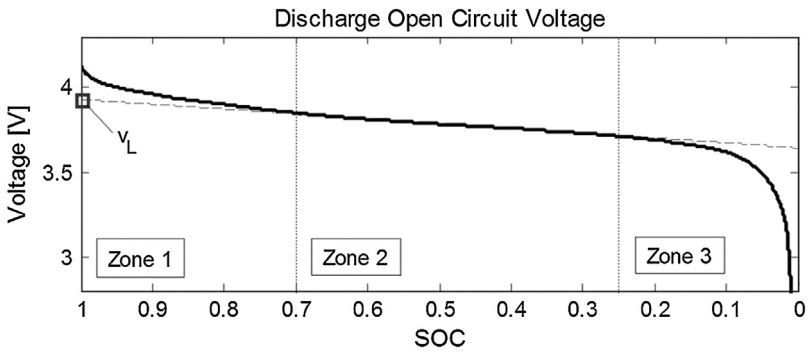


Fig. 1. OCV curve of a Li-Ion cell (black line) and the projection of its linear operational range (dashed gray line) as a function of SoC [5].

3.3. Acuña's time-of-failure probability mass function

Denoting healthy and faulty systems (at the kth time instant) by $\mathcal{H}_k$ and $\mathcal{F}_k$, respectively, the Acuña's ToF PMF [3] is defined as

$$\mathcal{P}(\mathcal{F}_k) = \mathcal{P}(\mathcal{F}_k|\mathcal{H}_{k_p:k-1}) \prod_{j=k_p+1}^{k-1} (1 - \mathcal{P}(\mathcal{F}_j|\mathcal{H}_{k_p:j-1})), \tag{35}$$

$\forall k > k_p$, where

$$\mathcal{P}(\mathcal{F}_k|\mathcal{H}_{k_p:k-1}) = \int_{\mathfrak{R}^{n_x}} p(failure|x_k) p(x_k|y_{1:k_p}) dx_k. \tag{36}$$

4. Case study: end-of-discharge time prognosis of lithium-ion batteries

We now proceed to apply the proposed methodology for prognostic algorithm design and hyper-parameter tuning on an illustrative case study, which corresponds to the problem of End-of-Discharge (EoD) time prognostics in lithium-ion (Li-Ion) batteries. This case study assumes that a filtering stage is carried out by a particle filtering algorithm, following the recommendations suggested in [5] (in terms of the number of particles utilized, among other implementation issues). In that regard, it is assumed that posterior estimates for both the battery State-of-Charge (SoC, defined as the ratio between the actual available energy and the maximum battery storage capacity E_{crit}) and internal polarization resistance are always available at the time where the prognostic algorithm is executed. The failure condition in this case is characterized by SoC levels below 10%.

4.1. State-space model

Filtering and prognostic stages use a state-space model to represent the evolution in time of the Li-Ion battery voltage as a function of (i) the SoC, (ii) the battery internal impedance, and (iii) the discharge current (exogenous system input). The objective in this case study is to prognosticate the moment in which the battery energy has depleted below 10%. As it has been already mentioned in Section 1, and as in any other prognostic problem, the "ground truth" failure PMF (in this case, the EoD time PMF) can be computed offline using Monte Carlo simulations for future trajectories of the state vector.

In actual implementations of failure prognostic algorithms, it is also necessary to characterize the future evolution of exogenous inputs to the state-space model (future operating profiles). Particularly, for EoD time prognostic purposes, [5] proposes to use a probabilistic characterization of the battery discharge current, via Markov Chains. However, it is important to note that, without loss of generality, both the performance assessment of a given prognostic algorithm and of the exogenous input characterization can be conducted separately [18]. Indeed, it is always possible to evaluate the performance of the prognostic algorithm conditional to a specific realization of the future usage profile, and then use the Law of Total Probability to incorporate the uncertainty associated with exogenous inputs. For this reason, hereafter, this study will assume that the future battery usage profile is known, solely focusing on computing the EoD time PMF conditional to that given profile.

For most of the battery operating range, the relationship between SoC and the *Open Circuit Voltage* (OCV) curve can be well characterized by an affine function; see "*zone 2*" in Fig. 1. However, we will use the state-space model proposed in [5], which allows to characterize the nonlinear behavior present in "*zone 1*" and "*zone 3*". Also, we have adopted a structure proposed in [19] to model the dependency between the polarization resistance and the battery discharge current.

State transition model

$$x_{k+1} = x_k - v_{oc}(x_k) \cdot u_k \cdot \frac{T_s}{E_{crit}} + \omega_k \tag{37}$$

Measurement model

$$y_k = v_{oc}(x_k) - u(k) \cdot R_{\text{int}}(x_k, u_k) + \eta_k, \tag{38}$$

with

$$v_{oc}(x_k) = \; v_L + (v_0 - v_L) \cdot e^{\gamma \cdot (x_2(k)-1)} + \alpha \cdot v_L \cdot (x_2(k) - 1) + (1 - \alpha) \cdot v_L \cdot (e^{-\beta} - e^{-\beta \cdot \sqrt{x_2(k)}}) \tag{39}$$

and

$$R_{\text{int}}(x_k, u_k) = r_0(u_k) + r_1(u_k) \cdot x_k + r_2(u_k) \cdot {x_k}^2. \tag{40}$$

In this representation, the input to the system $u_k = i_k[A]$ is defined as the discharge current, while $y_k = v_k[V]$ is the voltage at the battery terminals. The state x_k is the battery SoC measured with respect to E_{crit}, the expected total energy delivered by the battery; whereas the absolute value of the internal impedance is represented by the function $R_{int}(x_k, u_k)$. The process noise ω_k and the measurement noise η_k assume a zero mean Gaussian distribution. Finally, $T_s[s]$ is the sample time and v_0, v_L, α, β and γ are model parameters to be estimated offline (see [5] for more details).

Since a faulty condition is defined in this case by SoC values below a 10%, then Eq. (36) (required for computing the ToF PMF) becomes:

$$p(failure|x_k) = \mathbb{1}_{\{x \in \mathbb{R}: x < 0.1\}}(x_k). \tag{41}$$

4.2. Prognostic algorithm

A particle-filtering-based prognostic algorithm [4] is selected to illustrate the design methodology. This algorithm uses, as initial condition, an empirical state posterior distribution that results from a PF implementation. It also considers that prognostic stage begins at time k_p, and that the state posterior distribution at that time instant is denoted by

$$p(x_{k_p}|y_{1:k_p}) = \sum_{i=1}^{N_p} w_{k_p}^{(i)} \delta_{x_{k_p}^{(i)}}(x_{k_p}), \tag{42}$$

where N_p is the amount of samples used by the PF implementation.

(0) Resample $p(x_{k_p}|y_{1:k_p})$ and get a set of N_θ equally weighted particles.

Then, for each future time instant k, $k > k_p$, perform the following steps:

(1) Compute the expected state transitions $x_k^{*(i)} = \mathbb{E}\{f({x_{k-1}}^{(i)}, u_{k-1}, \omega_{k-1})\}$, $\forall i \in \{1, \ldots, N_\theta\}$, and calculate the empirical covariance matrix

$$\hat{S}_k = \frac{1}{N_\theta - 1} \sum_{i=1}^{N_\theta} [x_k^{*(i)} - \bar{x}_k^*][x_k^{*(i)} - \bar{x}_k^*]^T,$$

with $\bar{x}_k^* = \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} x_k^{*(i)}$.

(2) Compute $\hat{D}_k$ such that $\hat{D}_k \cdot \hat{D}_k^T = \hat{S}_k$.

(3) Update the samples as

$$x_k^{(i)} = x_k^{*(i)} + h_\theta \cdot \hat{D}_k \cdot \varepsilon_k^{(i)}, \quad \varepsilon_k^{(i)} \sim \mathcal{E},$$

where $\mathcal{E}$ is the Epanechnikov kernel and h_θ corresponds to its bandwidth.

Thus, in this case, the hyper-parameters vector for the prognostic algorithm is defined as $\theta^T = \begin{bmatrix} N_\theta & h_\theta \end{bmatrix}$.

4.3. Algorithm design: hyper-parameter tuning

The methodology presented in Section 3 is now utilized to tune hyper-parameters N_θ and h_θ of the prognostic algorithm proposed by [4] (see Section 4.2), when this algorithm is used to solve the problem of EoD time prognosis (see Section 4.1). Please note that the parameter N_θ directly affects the computational effort of the method (i.e.; $\theta_A = N_\theta$), while $\theta_B = h_\theta$ is more related to the capability of the algorithm to appropriately represent probability densities.

We will test the performance of the prognostic algorithm when predicting the EoD time at different moments during the battery discharge process. In this regard, and given that a full discharge takes approximately 11,628[s], we decided to execute the prognostic routine at 2000[s], 4000[s] and 6000[s] of operation. As it was previously mentioned in Section 4.1, the future discharge current is assumed to be known without loss of generality, since we aim at assessing the performance of the algorithms to characterize the evolution in time of the uncertainty associated with the state vector. The discharge current profile was generated from random realizations of a four-state Markov Chain. Each state of the Markov Chain is associated with a specific value for the battery discharge current: 5[A], 10[A], 15[A] and 20[A], respectively (see Fig. 2). It is assumed that transitions between these Markov Chain states occur every 10[s].

We now proceed to apply the proposed methodology to this case study, step-by-step.

4.3.1. Step 1: generate MCP-BCRLBs

The first step of the methodology is to compute the sequence of MCP- BCRLBs, which in turn requires computation of an initial condition for the recursion. The procedure to achieve this goal has been reported in [13], and the detailed equations for this particular case study can be found in Appendix C.1. The hyper-parameter N_θ is set to 100 particles, following the recommendations stated in [5]. As the recursion requires to compute expectations over predicted state probability density functions, the MCP-BCRLB cannot be computed analytically (see Appendix C.2 for detailed expressions). To overcome this difficulty, we simulate a million random trajectories for the evolution of the battery SoC using Eq. (37). Fig. 3 shows the results obtained when state predictions are computed at $k_p = 4000[s]$.

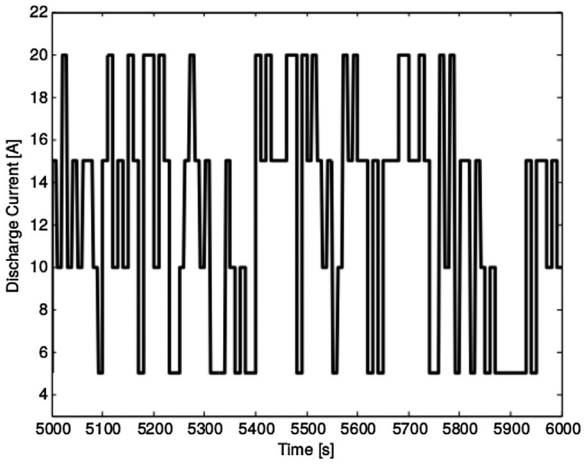
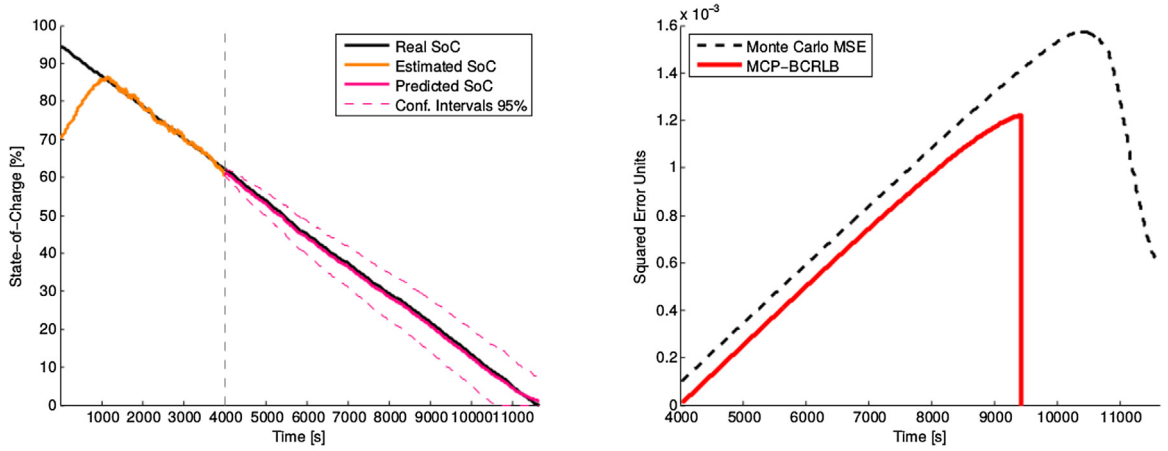


Fig. 2. Illustration of battery discharge current profile.



(a) SoC estimation and prognosis. (b) MSE and MCP-BCRLB curves.

Fig. 3. Example results for $k_p = 4000[s]$. Fig. 3(a) shows the results for battery SoC filtering and prediction stages. The estimation stage assumes an incorrect initial condition of 70% for the SoC, and is executed using a PF with 100 particles [5]. Long-term predictions are performed simulating a million random state trajectories. These predictions are used to compute MSE and MCP-BCRLB curves in Fig. 3(b).

A similar procedure can be utilized to compute the MCP-BCRLB when $k_p = 2000[s]$ or $k_p = 6000[s]$; see Fig. 4.

4.3.2. Step 2: choose candidates for algorithm hyper-parameters

The bandwidth h_θ of each Epanechnikov kernel has a theoretical optimal value h_{opt} when particles are sampled from Gaussian distribution with unity covariance matrix (see Eq. (43)) [4]. Although this is seldom the case, particle- filtering-based prognostic algorithm implementations use this value as an educated guess. Since in this case study $n_x = 1$, we should use $h_{opt} = 0.8529$.

$$h_{opt} = A \cdot N_\theta^{-\frac{1}{n_x+4}}, \quad A = (8 \cdot c_{n_x}^{-1} \cdot (n_x + 4) \cdot (2 \cdot \sqrt{\pi})^{n_x})^{\frac{1}{n_x+4}} \tag{43}$$

However, if we set $h_\theta = h_{opt}$ the implementation of the particle-filtering-based prognostic algorithm evidenced poor performance in terms of predicted state MSE (far greater than corresponding MCP- BCRLB). This fact motivates to search for other hyper-parameter candidates, possibly smaller that h_{opt}. For illustrative purposes, we will analyze the following options for the bandwidth for the Epanechnikov kernel: $h_{\theta,1} = 0.0048$, $h_{\theta,2} = 0.0046$, $h_{\theta,3} = 0.0044$, $h_{\theta,4} = 0.0042$ and $h_{\theta,5} = 0.0040$.

4.3.3. Step 3: discard hyper-parameter candidates related to implementations that violate MCP-BCRLBs

Fig. 5 shows the resulting predictive state MSE curves for realizations of the particle-filtering-based prognostic algorithm ($N_\theta = 100$) that used the proposed candidates for the hyper-parameter h_θ; all of them executed at time $k_p = 4000[s]$. It can be noticed that predictive MSE curves associated with candidates $h_{\theta,4}$ and $h_{\theta,5}$ violate the MCP-BCRLB curve. That is also the case for candidate $h_{\theta,2}$, although it must be noted that the bound is violated for a small set of time instants: a fact that may be caused by efficiency algorithm constraints that were originally imposed. It would be interesting to analyze how the algorithm performs with $h_{\theta,2}$ if N_θ increases. We will keep this idea in mind for Step 4).

The analysis is analogous for cases in which $k_p = 2000[s]$ and $k_p = 6000[s]$.

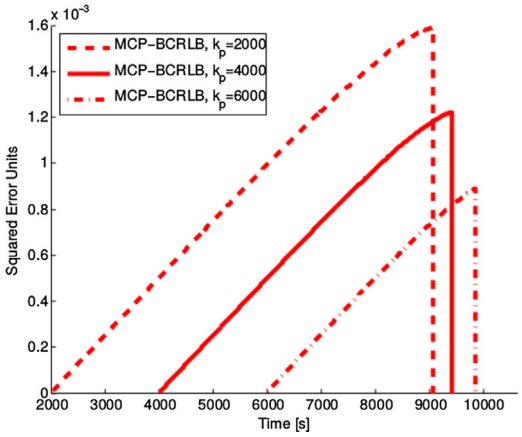


Fig. 4. MCP-BCRLB curves computed at different values of k_p.

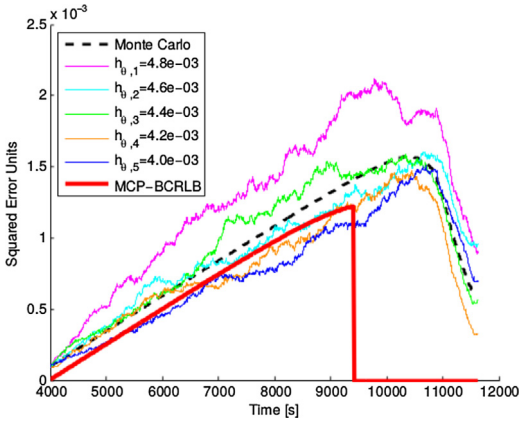


Fig. 5. Predictive state MSE and MCP- BCRLB curves computed at $k_p = 4000$ [s].

Table 1
Dissimilarity between predicted state MSE and MCP-BCRLB curves (ℓ^1 distance) for $N_\theta = 100$. Candidates that were discarded in Step 3 are marked with a $\times$ symbol. Chosen candidates are associated with minimum distances (marked with a $\surd$ symbol).

$\|\text{MSE}_{h_{\theta,i}} - \text{MCP-BCRLB}\|_1$	$h_{\theta,1}$	$h_{\theta,2}$	$h_{\theta,3}$	$h_{\theta,4}$	$h_{\theta,5}$
$k_p = 2000$ [s]	8.8529	6.4213	5.8543$\surd$	4.8402$\times$	6.0454$\times$
$k_p = 4000$ [s]	6.1874	3.5286$\times$	3.9176$\surd$	2.9338$\times$	3.1468$\times$
$k_p = 6000$ [s]	1.5946$\times$	1.6885$\surd$	2.0745$\times$	1.7413$\times$	1.6845$\times$

4.3.4. Step 4: use the ℓ^1-norm to select the most appropriate hyper-parameter candidate

After executing Step 3, two candidates remain: $h_{\theta,1}$ and $h_{\theta,3}$. Although Fig. 5 suggests that $h_{\theta,3}$ may have better performance, it is important to select the value of the hyper-parameter using a rigorous criteria: we propose to use the ℓ^1-norm to measure the distance between predicted state MSEs and MCP-BCRLB curves:

$$\|\text{MSE}_{h_{\theta,1}} - \text{MCP-BCRLB}\|_1 = 6.1874$$

$$\|\text{MSE}_{h_{\theta,3}} - \text{MCP-BCRLB}\|_1 = 3.9176.$$

Following an analogous procedure for cases in which $k_p = 2000[s]$ and $k_p = 6000[s]$, it is possible to assign adequate values of θ as a function of the elapsed time. Results are summarized in Table 1.

It is important to note that candidate $h_{\theta,3} = 0.0044$ represents an appropriate choice if the prediction horizon is relatively large. However, as we approach the battery discharge time, only $h_{\theta,2} = 0.0046$ becomes a feasible option. This result is strongly conditioned to the number of particles used in the PF-based prognostic algorithm implementation ($N_\theta = 100$). Intuition indicates that it is worthwhile to explore which could be the best choice for h_θ if we relax efficiency constraints and allow a larger number of particles in the implementation.

4.3.5. Step 5: relax efficiency soft-constraints

According to the proposed methodology, we now proceed to relax soft- constraints associated with efficiency criteria (i.e., the number of particles N_θ). This procedure will help us to understand the cost associated with computational effort constraints, measured in terms of the efficacy of the prognostic algorithm. In this regard, let us increase the hyper- parameter value to $N_\theta = 500$. After going through Steps

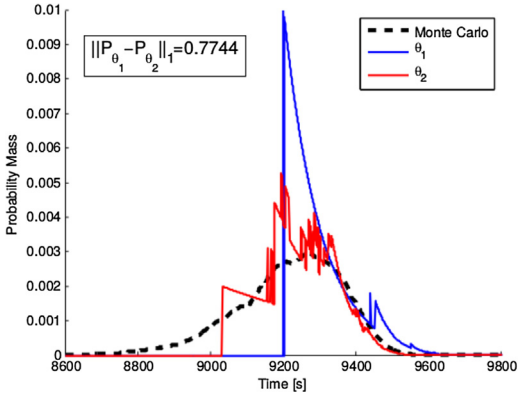


Fig. 6. Time-of-Failure PMFs for two choices of algorithm hyper-parameters $\theta_1^T = [100 \quad 0.0044]$ and $\theta_2^T = [500 \quad 0.0044]$. The latter choice, θ_2, assumes a relaxation of efficiency constraints (i.e., $N_\theta = 500$). Prognosis is executed at time $k_p = 4000[s]$, and EoD PMFs are computed using solely one realization of the probability-based algorithm. Black-dashed line shows the "ground truth" EoD PMF, which is approximated by a million simulations of the state transition model (37) .

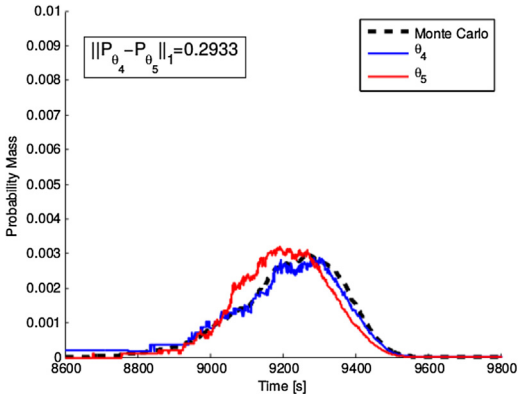


Fig. 7. Time-of-Failure PMFs for two choices of algorithm hyper-parameters $\theta_4^T = [5000 \quad 0.0044]$ and $\theta_5^T = [10,000 \quad 0.0044]$. Prognosis is executed at time $k_p = 4000$ [s], and EoD PMFs are computed using solely one realization of the probability-based algorithm. Black-dashed line shows the "ground truth" EoD PMF, which is approximated by a million simulations of the state transition model (37).

1–4 once more, it is interesting to note that in this new hypothetical scenario, the most appropriate hyper-parameters vector choice would have been different (see Table D.2 in the Appendix for more details):

$$\theta^T = \begin{bmatrix} N_\theta & h_{\theta,2} \end{bmatrix} = \begin{bmatrix} 500 & 0.0044 \end{bmatrix}.$$

Although the aforementioned steps allow to choose hyper-parameter candidates in terms of the resulting predicted state MSE, we should also assess the impact in terms of the true outcome of probability-based prognostic algorithms: the ToF PMF. For this purpose, we need to use a metric that could help to characterize differences between two (or more) ToF PMFs. Using this metric, the designer would be able to avoid decreasing the efficiency of the implementation if the associated increment on its efficacy is negligible. In this regard, we suggest the use of an ℓ^1 distance between discrete-time functions as a measure of changes between two ToF PMFs. Indeed, let us consider the case illustrated in Fig. 6, which shows the ToF PMFs obtained when implementing a PF-based algorithm for the EoD time prognostics with two choices for the hyper-parameters vector: $\theta_1^T = [100 \quad 0.0044]$ and $\theta_2^T = [500 \quad 0.0044]$. The latter choice, θ_2, assumes a relaxation of efficiency constraints (i.e., $N_\theta = 500$ instead of 100 particles). The ℓ^1 distance between these two PMFs indicate that the impact associated with the increment in the size of the particle population is not negligible. Furthermore, a direct comparison with the "ground truth" EoD PMF, approximated by a million simulations of the state transition model, indicates that this increment in N_θ allows a better characterization of the left tail of the ToF PMF.

Thus, we may now proceed to explore the impact of a larger increase in the number of particles N_θ, in terms of the achievable limits for algorithm efficacy. Consider, for this purpose, the following sequence of hyper-parameters candidates, indexed according to an efficiency criterion:

$$\begin{aligned} \{\theta_n\}_{n=1}^5 &= \left\{ \begin{bmatrix} N_{\theta_n} \\ h_{\theta_n} \end{bmatrix} \right\}_{n=1}^5 \\ &= \left\{ \begin{bmatrix} 100 \\ 0.0044 \end{bmatrix}, \begin{bmatrix} 500 \\ 0.0044 \end{bmatrix}, \begin{bmatrix} 1000 \\ 0.0042 \end{bmatrix}, \begin{bmatrix} 5000 \\ 0.0044 \end{bmatrix}, \begin{bmatrix} 10,000 \\ 0.0044 \end{bmatrix} \right\}. \end{aligned}$$

Fig. 7 illustrates the ToF PMFs obtained when using the two last candidates of the proposed sequence in the implementation of the PF-based prognostic algorithm: $\theta_4^T = [5000 \quad 0.0044]$ and $\theta_5^T = [10,000 \quad 0.0044]$. It is interesting to note that differences in terms of the characterization of the left tail of the PMF (the most useful source of information to quantify operational risk in prognostics) is small compared to the increment in computational complexity. Moreover, when comparing with respect to the "ground truth" EoD PMF,

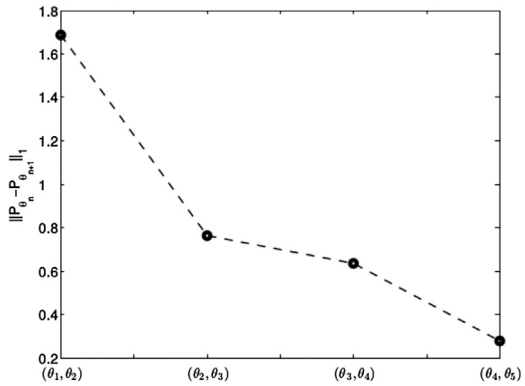


Fig. 8. Summary of ℓ^1 distances between EoD PMFs using single realizations of the probability-based algorithm on each iteration. The nth iteration of the design procedure is related to hyper-parameters vector θ_n. Prognostics are executed at time $k_p = 4000$ [s].

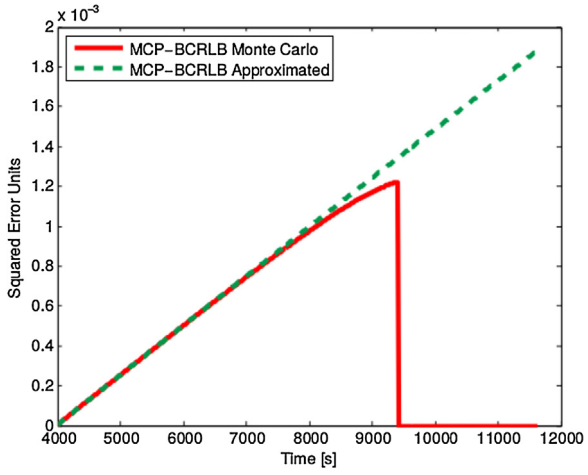


Fig. 9. Comparison of obtained MCP- BCRLBs. The red thick line corresponds to a bound computed by simulating a million state trajectories. Green dashed line uses the proposed analytic solution in an approximated state model.

evidence indicates that efficacy does not increase when $N_\theta = 10,000$. Thus, in terms of the final design for this specific case study, we would recommend to use $\theta_4^T = [5000 \quad 0.0044]$. This final hyper- parameter choice aims at a combination that provides reasonable results in terms of a truthful characterization for the risk of failure, using the least computational resources.

To complement the previous analysis, Fig. 8 depicts the evolution of ℓ^1 distances between obtained ToF PMFs as we proceed comparing the performance associated with candidates $\theta_1, \ldots, \theta_5$ (where θ_n is the candidate selected in the nth iteration of the design procedure). It is interesting to note that the relaxation of efficiency constrains (N_θ, in this case) entails strictly decreasing differences regarding EoD PMFs results in terms of the proposed metric (ℓ^1 distance between ToF PMFs).

Further information about the performance exhibited by the prognostic algorithm when using other elements of the sequence of candidates can be found in Tables D.2–Table D.5; see Appendix.

4.4. Avoiding Monte Carlo simulations in EoD prognostic algorithms

The battery discharge model can be easily approximated by a structure that holds the necessary requirements to obtain an analytic expression of MCP-BCRLBs (see Section 3.2). The state transition equation for the SoC is:

$$x_{k+1} = x_k - v_{oc}(x_k) \cdot u_k \cdot \frac{T_s}{E_{crit}} + \omega_k,$$

which is nonlinear with respect to x_k because of the term $v_{oc}(x_k)$. However, from Fig. 1 we can recognize a wide operating zone in which $v_{oc}(x_k)$ is linear with respect to x_k. Indeed, if we linearize $v_{oc}(x_k)$ around $x_o = 0.5$, we obtain:

$$v_{oc}(x_k) \approx v_{oc}(x_o) + \left.\frac{\partial v_{oc}(x_k)}{\partial x_k}\right|_{x_k = x_o} \cdot (x_k - x_o)$$

And thus, the state transition equation can be approximated by:

$$x_{k+1} = \underbrace{(1 - \frac{\partial v_{oc}(x_k)}{\partial x_k}|_{x_k=x_o} \cdot u_k \cdot \frac{T_s}{E_{crit}})}_{A_k(u_k)} \cdot x_k + \underbrace{(\frac{\partial v_{oc}(x_k)}{\partial x_k}|_{x_k=x_o} \cdot x_o - v_{oc}(x_o)) \cdot u_k \cdot \frac{T_s}{E_{crit}}}_{B_k(u_k)} + \omega_k,$$

expression that has the required form $x_{k+1} = A_k(u_k) \cdot x_k + B_k(u_k) + \omega_k$.

Fig. 9 shows a comparison between the MCP-BCRLB curve that is obtained with the proposed approximation and the one that results from the computation of Monte Carlo simulation of the battery discharge model. It is noteworthy that design methodology would have led to the same conclusions if the prediction horizon is limited to 4000[s] of operation. This would have resulted in a prognostic algorithm that would have been able to perform properly with prediction windows that exceeded an hour of anticipation.

5. Conclusions

This article presents a novel prognostic performance metric based on the concept of Bayesian Cramér-Rao Lower Bounds (BCRLBs) for the predicted state mean square error (MSE), which is conditional to measurement data and model dynamics. This metric allows to implement a formal step-by-step design methodology to tune prognostic algorithm hyper-parameters, which allows to guarantee that obtained results do not violate fundamental precision bounds. Both the metric and the proposed design methodology are verified and validated using the problem of End-of-Discharge time prognosis as a case study. In this regard, it is safe to state that the contributions of this research effort are both theoretical and practical, with special focus on probability-based prognostic algorithms.

The design methodology distinguishes between hyper-parameters that affect the efficiency of the implementation and those that have impact on the efficacy of obtained results, providing a structured procedure to explore different combinations that could improve the characterization of the ToF PMF. We demonstrated that the proposed design procedure allows to detect situations in which the prognostic algorithm implementation generates results with greater precision than it is possible to achieve. Furthermore, it allows to measure the impact of a relaxation in efficiency constraints on the outcome of the prognostic algorithm; thus helping the designer to take an informed decision on the hardware that is required to implement the algorithm for real-time applications.

Finally, it is important to mention that, for the very first time, it was possible to tune the parameters of a PF-based prognostic algorithm with regularization of the predicted state probability density using a formal and structured procedure. This outcome will undoubtedly be of value for members of the PHM community who are currently using this algorithm in Li-Ion battery EoD time prognostic problems, or in other challenges of the same nature.

Acknowledgements

This work has been supported by FONDECYT Chile Grant No. 1170044, and the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, CONICYT. Also, the authors want to thank project CONICYT PIA ACT1405 and CONICYT-PFCHA/Doctorado Nacional/2017-21171097 for supporting David E. Acuña doctoral studies.

Appendix A. Proof of Theorem 3.1

Let $\hat{x}^i = \hat{x}^i(y_{1:k_p})$ be an estimator of x^i. Since p_k^{cp} is absolutely continuous, $\frac{\partial p_k^{cp}}{\partial x^i}$ is well defined, moreover, by hypothesis, it is absolutely integrable. Therefore, we have the following identity

$$\begin{aligned}
&\int_{\mathbb{R}^{(k-k_p+1)n_X}} \frac{\partial p_k^{cp}}{\partial x^i}(x_{k_p:k}) dx_{k_p:k} \\
&= \int_{\mathbb{R}^{(k-k_p+1)(n_X-1)}} \lim_{N\to\infty} \int_{[-N,N]} \cdots \int_{[-N,N]} \frac{\partial p_k^{cp}}{\partial x^i}(x_{k_p:k\setminus i}, x^i_{k_p}, \ldots x^i_k) dx^i_{k_p} \ldots dx^i_k dx_{k_p:k\setminus i} \\
&= \int_{\mathbb{R}^{(k-k_p+1)(n_X-1)}} (p_k^{cp}(x_{k_p:k\setminus i}, x^i))\Big|_{-\infty}^{+\infty} dx_{k_p:k\setminus i} \\
&= 0.
\end{aligned}$$

where the last identity is implied by the hypothesis of

$$\lim_{x^i\to+\infty} x^i p(x_{k_p:k}) = \lim_{x^i\to-\infty} x^i p(x_{k_p:k}) = 0$$

Then, simply multiplying by $\hat{x}^i(y_{1:k_p})$ we get

$$\int \hat{x}^i(y_{1:k_p}) \frac{\partial p_k^{cp}}{\partial x^i}(x_{k_p:k}) dx_{k_p:k} = 0. \tag{A.1}$$

On the other hand, integrating by parts one gets that

$$\int_{-\infty}^{+\infty} x^i \frac{\partial p_k^{cp}}{\partial x^i} dx^i = (x^i p_k^{cp})\Big|_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} p_k^{cp} dx^i.$$

Due to condition 2 (see Eq. (28)), $(x^i p_k^{cp})\big|_{-\infty}^{+\infty} = 0$. Integrating with respect to $x_{k_p:k\setminus i}$ ($x_{k_p:k}$ omitting the i-th element), then

$$\int x^i \frac{\partial p_k^{cp}}{\partial x^i} dx_{k_p:k} = -1. \tag{A.2}$$

Subtracting Eq. (A.2) to Eq. (A.1), we obtain, $\forall y_{1:k_p}$,

$$\int (\hat{x}^i(y_{1:k_p}) - x^i) \frac{\partial p_k^{cp}}{\partial x^i} dx_{k_p:k} = \int (\hat{x}^i(y_{1:k_p}) - x^i) \frac{\partial \log p_k^{cp}}{\partial x^i} p_k^{cp} dx_{k_p:k} = 1. \tag{A.3}$$

Similarly for $j \in \{1, 2, \ldots, (k-k_p+1)\, n_x\} \setminus \{i\}$, we have $\forall y_{1:k_p}$

$$\int (\hat{x}^i(y_{1:k_p}) - x^i) \frac{\partial \log p_k^{cp}}{\partial x^j} p_k^{cp} dx_{k_p:k} = 0. \tag{A.4}$$

Combining Eqs. (A.3) and (A.4) in a matrix form, we get

$$\int (\hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k})[\nabla_{x_{k_p:k}} \log p_k^{cp}] p_k^{cp} dx_{k_p:k} = I_{(k-k_p+1)n_x},$$

where $I_{(k-k_p+1)n_x}$ is the $(k-k_p+1)\, n_x$-dimensional identity matrix. Pre-multiplying and post-multiplying the last equation by a^T and b, $a, b \in \mathbb{R}^{(k-k_p+1)n_x}$, respectively, we have

$$\begin{aligned} a^T b &= \int a^T (\hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k})[\nabla_{x_{k_p:k}} \log p_k^{cp}] b p_k^{cp} dx_{k_p:k} \\ &= \int a^T (\hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k}) \sqrt{p_k^{cp}} [\nabla_{x_{k_p:k}} \log p_k^{cp}] b \sqrt{p_k^{cp}} dx_{k_p:k}. \end{aligned}$$

Applying Cauchy–Schwarz inequality

$$\begin{aligned} (a^T b)^2 \quad \leq \quad & \left(\int a^T (\hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k})(\hat{x}_{k_p:k}(y_{1:k_p}) - x_{k_p:k})^T a p_k^{cp} dx_{k_p:k} \right) \ldots \\ & \ldots \cdot \left(\int [\nabla_{x_{k_p:k}} \log p_k^{cp}] b b^T [\nabla_{x_{k_p:k}} \log p_k^{cp}]^T p_k^{cp} dx_{k_p:k} \right). \end{aligned}$$

Expressing this in terms of conditional expectation, we get the following

$$\begin{aligned} (a^T b)^2 \quad \leq \quad & (a^T \mathbb{E}_{p_k^{cp}} \{\tilde{x}_{k_p:k} \tilde{x}_{k_p:k}^T | y_{1:k_p}\} a) \ldots \\ & \ldots \cdot \left(\int [\nabla_{x_{k_p:k}} \log p_k^{cp}] b b^T [\nabla_{x_{k_p:k}} \log p_k^{cp}]^T p_k^{cp} dx_{k_p:k} \right)^T \\ = \quad & (a^T \mathbb{E}_{p_k^{cp}} \{\tilde{x}_{k_p:k} \tilde{x}_{k_p:k}^T | y_{1:k_p}\} a) \ldots \\ & \ldots \cdot (b^T \mathbb{E}_{p_k^{cp}} \{[\nabla_{x_{k_p:k}}]^T \log p_k^{cp}][\nabla_{x_{k_p:k}} \log p_k^{cp}]\} b). \end{aligned}$$

Defining $I_{cp}(x_{k_p:k}|y_{1:k_p}) \triangleq \mathbb{E}_{p_k^{cp}} \{[\nabla_{x_{k_p:k}}]^T \log p_k^{cp}][\nabla_{x_{k_p:k}} \log p_k^{cp}]\}$ and choosing $b = I_{cp}^{-1}(x_{k_p:k}|y_{1:k_p}) a$, we obtain

$$a^T (\mathbb{E}_{p_k^{cp}} \{\tilde{x}_{k_p:k} \tilde{x}_{k_p:k}^T | y_{1:k_p}\} - I_{cp}^{-1}(x_{k_p:k}|y_{1:k_p})) a \geq 0.$$

Given that $a \in \mathbb{R}^{(k-k_p+1)n_x}$ is arbitrary, $\mathbb{E}_{p_k^{cp}} \{\tilde{x}_{k_p:k} \tilde{x}_{k_p:k}^T | y_{1:k_p}\} - I_{cp}^{-1}(x_{k_p:k}|y_{1:k_p})$ must necessarily be a semi-definite positive matrix.

Appendix B. Proof of Theorem 3.2

We need some previous results before demonstrating the Theorem 3.2, which were extracted from [20].

Theorem Appendix B.1. *Defining the score function $s(\theta, z) = \nabla_\theta \log p(\theta|z)$ and taking $t(\cdot,\cdot)$ a vectorial function of z and θ with values in $\mathbb{R}^{n_\theta}$, and assuming some regularity, the following identity holds*

$$\mathbb{E}_{p(\theta|z)} \{t(\theta, z) s(\theta, z)\} = \nabla_\theta \mathbb{E}_{p(\theta|z)} \{t(\theta, z)\} - \mathbb{E}_{p(\theta|z)} \{\nabla_\theta t(\theta, z)\}. \tag{B.1}$$

Proof. We have

$$\mathbb{E}_{p(\theta|z)} \{t(\theta, z)\} = \int t(\theta, z) p(\theta|z) dz.$$

Applying gradient operator at both sides and assuming that differentiation conditions are fulfilled so that the gradient can get into the integral:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p(\theta|z)}\{t(\theta,z)\} &= \nabla_\theta \int t(\theta,z)p(\theta|z)dz \\ &= \int [\nabla_\theta t(\theta,z)p(\theta|z)]dz \\ &= \int t(\theta,z)[\nabla_\theta p(\theta|z)]dz + \int [\nabla_\theta t(\theta,z)]p(\theta|z)dz \\ &= \int t(\theta,z)[\nabla_\theta \log p(\theta|z)]p(\theta|z)dz + \int [\nabla_\theta t(\theta,z)]p(\theta|z)dz \\ &= \mathbb{E}_{p(\theta|z)}\{t(\theta,z)s(\theta,z)\} + \mathbb{E}_{p(\theta|z)}\{\nabla_\theta t(\theta,z)\}\end{aligned}$$

□

Corollary 1. *If $s(\theta,z)$ is the score function of a differentiable likelihood $p(\theta|z)$, then*

$$\mathbb{E}_{p(\theta|z)}\{s(\theta,z)\} = 0.$$

Proof. By Theorem B.1, and because of Eq. (B.1), for each constant vector t we have that:

$$\begin{aligned}t\mathbb{E}_{p(\theta|z)}\{s(\theta,z)\} &= \mathbb{E}_{p(\theta|z)}\{ts(\theta,z)\} \\ &= \nabla_\theta \mathbb{E}_{p(\theta|z)}\{t\} - \mathbb{E}_{p(\theta|z)}\{\nabla_\theta t\} \\ &= 0.\end{aligned}$$

As the latter expression is valid for all t, it follows that $\mathbb{E}_{p(z|\theta)}\{s(\theta,z)\} = 0$.
□

Lemma Appendix B.2. *If the score function $s(y_{1:k_p}, x_{k_p:k}) = \nabla_{x_{k_p:k}} \log p(x_k|y_{1:k_p})$ is differentiable, then $I_{cp}(x_{k_p:k}|y_{1:k_p})$ can be expressed as*

$$I_{cp}(x_{k_p:k}|y_{1:k_p}) = \mathbb{E}_{p_k^{cp}}\{-\Delta^{x_{k_p:k}}_{x_{k_p:k}} \log p_k^{cp}\}. \tag{B.2}$$

Proof. By Theorem B.1 and Corollary 1, substituting $\theta = x_{k_p:k}$ and $z = y_{1:k_p}$, we have

$$\mathbb{E}_{p_k^{cp}}\{t(y_{1:k_p}, x_{k_p:k})s(y_{1:k_p}, x_{k_p:k})\} = \nabla_{x_{k_p:k}} \mathbb{E}_{p_k^{cp}}\{t(y_{1:k_p}, x_{k_p:k})\} - \mathbb{E}_{p_k^{cp}}\{\nabla_{x_{k_p:k}} t(y_{1:k_p}, x_{k_p:k})\}.$$

Taking the function $t(y_{1:k_p}, x_{k_p:k}) = s(y_{1:k_p}, x_{k_p:k})^T$ and using the previous result, we get

$$\mathbb{E}_{p_k^{cp}}\{t(y_{1:k_p}, x_{k_p:k})s(y_{1:k_p}, x_{k_p:k})\} = -\mathbb{E}_{p_k^{cp}}\{\nabla_{x_{k_p:k}} t(y_{1:k_p}, x_{k_p:k})\},$$

and the result is straightforward.□
Taking into account these previous results, we are now able to prove Theorem 3.2:
We have

$$\log p(x_{k_p:k}|y_{1:k_p}) = \log p(x_{k_p}|y_{1:k_p}) + \sum_{i=k_p+1}^{k} \log p(x_i|x_{i-1}).$$

Considering that $D_i = S_i^i + S_{i+1}^i$, $I_{cp}(x_{k_p:k}|y_{1:k_p})$ can be decomposed into a matrix of four blocks in the following manner

$$I_{cp}(x_{k_p:k}|y_{1:k_p}) = \left[\begin{array}{ccc|c} D_{k_p} & S_{k_p+1}^{k_p,k_p+1} & & \\ S_{k_p+1}^{k_p+1,k_p} & \ddots & \ddots & \\ & \ddots & D_{k-1} & S_k^{k-1,k} \\ \hline & & S_k^{k,k-1} & S_k^k \end{array}\right] = \left[\begin{array}{c|c} J_k^{11} & J_k^{12} \\ \hline J_k^{21} & J_k^{22} \end{array}\right],$$

where empty spaces represent zeros. Considering the previous definitions, it can be verified that

$$I_{cp}(x_{k_p:k}|y_{1:k_p}) = \left[\begin{array}{cc|c} J_{k-1}^{11} & J_{k-1}^{12} & \\ J_{k-1}^{21} & J_{k-1}^{22} + S_k^{k-1} & S_k^{k-1,k} \\ \hline & S_k^{k,k-1} & S_k^k \end{array}\right] \tag{B.3}$$

On the other hand, we have

$$\mathbb{E}_{p_k^{cp}}\{\tilde{x}_{k_p:k}\tilde{x}_{k_p:k}^T|y_{1:k_p}\} \geq \left[\begin{array}{c|c} J_k^{11} & J_k^{12} \\ \hline J_k^{21} & J_k^{22} \end{array}\right]^{-1} = \left[\begin{array}{c|c} C_k^{11} & C_k^{12} \\ \hline C_k^{21} & C_k^{22} \end{array}\right] \tag{B.4}$$

Then, the MSE associated to any estimator of x_k is lower bounded by

$$\mathbb{E}_{p_k^{cp}}\{\tilde{x}_k\tilde{x}_k^T|y_{1:k_p}\} \geq C_k^{22}.$$

Taking into account Eqs. (B.3) and (B.4), and applying the matrix inversion rule,

$$\begin{aligned}[C_k^{22}]^{-1} &= J_k^{22} - J_k^{21}[J_k^{11}]^{-1}J_k^{12} \\ &= S_k^k - S_k^{k,k-1}[J_{k-1}^{22} + S_k^{k-1} - J_{k-1}^{21}[J_{k-1}^{11}]^{-1}J_{k-1}^{12}]^{-1}S_k^{k-1,k} \\ &= S_k^k - S_k^{k,k-1}[[C_{k-1}^{22}]^{-1} + S_k^{k-1}]^{-1}S_k^{k-1,k}\end{aligned}$$

with the initial condition $[C_{k_p}^{22}]^{-1} = S_{k_p}^{k_p} = \mathbb{E}\{-\Delta_{x_{k_p}^{k_p}}^{x_{k_p}} \log p(x_{k_p}|y_{1:k_p})\}$.

Appendix C. MCP-BRCLB initial condition and recursion elements

C.1 Initial condition

Below we develop a series of equations for obtaining the initial condition for the computation of the MCP-BCRLB recursive sequence. We aim at computing:

$$[C_{k_p}^{22}] = \mathbb{E}\{-\Delta_{x_{k_p}^{k_p}}^{x_{k_p}} \log p(x_{k_p}|y_{1:k_p})\}^{-1}.$$

Following the indications reported in [13], we have that

$$\begin{aligned}p(x_k|y_{1:k}) &= \frac{p(x_k, y_k, y_{1:k-1})}{p(y_k, y_{1:k-1})} \\ &= \frac{p(y_k|x_k, y_{1:k-1})p(x_k|y_{1:k-1})p(y_{1:k-1})}{p(y_k|y_{1:k-1})p(y_{1:k-1})} \\ &= \frac{\overbrace{p(y_k|x_k)}^{likelihood}\overbrace{p(x_k|y_{1:k-1})}^{prior}}{\underbrace{p(y_k|y_{1:k-1})}_{evidence}}\end{aligned}$$

Now, we have

$$-\log p(x_k|y_{1:k}) = -\log p(y_k|x_k) - \log p(x_k|y_{1:k-1}) + \log p(y_k|y_{1:k-1})$$

It is required to get second derivatives with respect to x_k before applying expectation. Let's proceed term by term:

- **Likelihood**:

$$\begin{aligned}-\log p(y_k|x_k) &= -\log \frac{1}{\sqrt{2\pi}\sigma_v} e^{-\frac{1}{2}\frac{(y_k - (v_{oc}(x_k) - I_k R_{\text{int}}(x_k, I_k)))^2}{\sigma_v^2}} \\ &= c_0 + \frac{1}{2}\frac{(y_k - (v_{oc}(x_k) - I_k R_{\text{int}}(x_k, I_k)))^2}{\sigma_v^2}\end{aligned}$$

$$\begin{aligned}\Rightarrow -\frac{\partial \log p(y_k|x_k)}{\partial x_k} &= -\left(\frac{y_k - (v_{oc}(x_k) - I_k R_{\text{int}}(x_k, I_k))}{\sigma_v^2}\right)\cdots \\ &\quad \cdots\cdot\left(\frac{\partial v_{oc}(x_k)}{\partial x_k} - I_k\frac{\partial R_{\text{int}}(x_k, I_k)}{\partial x_k}\right)\end{aligned}$$

$$\begin{aligned}\Rightarrow -\frac{\partial^2 \log p(y_k|x_k)}{\partial {x_k}^2} &= \frac{1}{\sigma_v^2}\left(\frac{\partial v_{oc}(x_k)}{\partial x_k} - I_k\frac{\partial R_{\text{int}}(x_k, I_k)}{\partial x_k}\right)^2\cdots \\ &\quad \cdots - \left(\frac{y_k - (v_{oc}(x_k) - I_k R_{\text{int}}(x_k, I_k))}{\sigma_v^2}\right)\cdots \\ &\quad \cdots\cdot\left(\frac{\partial^2 v_{oc}(x_k)}{\partial {x_k}^2} - I_k\frac{\partial^2 R_{\text{int}}(x_k, I_k)}{\partial {x_k}^2}\right)\end{aligned}$$

- **Prior**: Firstly, note that

$$\begin{aligned} p(x_k|y_{1:k-1}) &= \int_{\mathcal{X}_{k-1}} p(x_k, x_{k-1}|y_{1:k-1})dx_{k-1} \\ &= \int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1}, y_{1:k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \\ &= \int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1} \end{aligned} \tag{C.1}$$

Then, since $p(\cdot|x_{k-1})$ is sufficiently regular,

$$-\log p(x_k|y_{1:k-1}) = -\log\left(\int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}\right)$$

$$\begin{aligned} \Rightarrow -\frac{\partial \log p(x_k|y_{1:k-1})}{\partial x_k} &= -\frac{1}{p(x_k|y_{1:k-1})}\frac{\partial p(x_k|y_{1:k-1})}{\partial x_k} \\ &= -\frac{\displaystyle\int_{\mathcal{X}_{k-1}} \frac{\partial p(x_k|x_{k-1})}{\partial x_k} p(x_{k-1}|y_{1:k-1})dx_{k-1}}{\displaystyle\int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}} \end{aligned}$$

$$\begin{aligned} \Rightarrow -\frac{\partial^2 \log p(x_k|y_{1:k-1})}{\partial {x_k}^2} &= -\frac{\displaystyle\int_{\mathcal{X}_{k-1}} \frac{\partial^2 p(x_k|x_{k-1})}{\partial {x_k}^2} p(x_{k-1}|y_{1:k-1})dx_{k-1}}{\displaystyle\int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}} \cdots \\ &\quad \cdots + \left(\frac{\displaystyle\int_{\mathcal{X}_{k-1}} \frac{\partial p(x_k|x_{k-1})}{\partial x_k} p(x_{k-1}|y_{1:k-1})dx_{k-1}}{\displaystyle\int_{\mathcal{X}_{k-1}} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}}\right)^2 \end{aligned}$$

On the other hand

$$p(x_k|x_{k-1}) = \frac{1}{\sqrt{2\pi}\sigma_\omega} e^{-\frac{1}{2}\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2}}$$

$$\begin{aligned} \Rightarrow \frac{\partial p(x_k|x_{k-1})}{\partial x_k} = &-\frac{1}{\sqrt{2\pi}\sigma_\omega^3} e^{-\frac{1}{2}\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2}} \cdots \\ &\cdots \cdot (x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}})) \end{aligned}$$

$$\Rightarrow \frac{\partial^2 p(x_k|x_{k-1})}{\partial {x_k}^2} = \frac{1}{\sqrt{2\pi}\sigma_\omega^5} e^{-\frac{1}{2}\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2}} \cdots$$
$$\cdots \cdot (x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2 \cdots$$
$$\cdots - \frac{1}{\sqrt{2\pi}\sigma_\omega^3} e^{-\frac{1}{2}\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2}}$$
$$= \frac{1}{\sqrt{2\pi}\sigma_\omega^3} e^{-\frac{1}{2}\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2}} \cdots$$
$$\cdots \cdot \left(\frac{(x_k-(x_{k-1}-I_{k-1}v_{oc}(x_{k-1})\frac{T_s}{E_{crit}}))^2}{\sigma_\omega^2} - 1 \right)$$

- **Evidence:** Since $p(y_k|\, y_{1:k-1})$ does not depend on x_k, it follows that

$$\Rightarrow \frac{\partial^2 \log p(y_k|y_{1:k-1})}{\partial {x_k}^2} = 0$$

Therefore, the MCP-BCRLB at the prognostic time instant k_p can be approximated considering the state posterior distributions at times k_p and $k_p - 1$, which are of the form:

$$p(x_{k_p}|y_{1:k_p}) \approx \sum_{i=1}^{N_p} w_{k_p}^{(i)} \delta_{x_{k_p}^{(i)}}(x_{k_p})$$

$$p(x_{k_p-1}|y_{1:k_p-1}) \approx \sum_{i=1}^{N_p} w_{k_p-1}^{(i)} \delta_{x_{k_p-1}^{(i)}}(x_{k_p-1})$$

Nonetheless, all these calculations require the following equations:

$$x_{k+1} = x_k - I_k v_{oc}(x_k)\frac{T_s}{E_{crit}} + \omega_k$$

$$y_k = v_{oc}(x_k) - I_k R_{\text{int}}(x_k, I_k) + \nu_k$$

$$v_{oc}(x_k) = v_L + (v_0 - v_L)\cdot e^{\gamma\cdot(x_k-1)} + \alpha\cdot v_L\cdot(x_k-1) + (1-\alpha)\cdot v_L\cdot(e^{-\beta} - e^{-\beta\cdot\sqrt{x_k}})$$

$$\frac{\partial v_{oc}(x_k)}{\partial x_k} = (v_0 - v_L)\cdot\gamma\cdot e^{\gamma\cdot(x_k-1)} + \alpha\cdot v_L + (1-\alpha)\cdot v_L\cdot e^{-\beta\cdot\sqrt{x_k}}\cdot\frac{\beta}{2\sqrt{x_k}}$$

$$\frac{\partial^2 v_{oc}(x_k)}{\partial {x_k}^2} = (v_0 - v_L)\cdot\gamma^2\cdot e^{\gamma\cdot(x_k-1)} - (1-\alpha)\cdot v_L\cdot e^{-\beta\cdot\sqrt{x_k}}\cdot\frac{\beta}{4}\cdot\left(\frac{\beta}{x_k} + \frac{1}{{x_k}^{3/2}}\right)$$

$$R_{\text{int}}(x_k, I_k) = p_0(I_k) + p_1(I_k)x_k + p_2(I_k)x_k^2$$

$$\frac{\partial R_{\text{int}}(x_k, I_k)}{\partial x_k} = p_1(I_k) + 2p_2(I_k)x_k$$

$$\frac{\partial^2 R_{\text{int}}(x_k, I_k)}{\partial {x_k}^2} = 2p_2(I_k)$$

C.2 Elements of the recursion

$$-\log p(x_{k+1}|x_k) = -\log \frac{1}{\sqrt{2\pi}\sigma_\omega} e^{-\frac{1}{2}\frac{\left(x_{k+1}-\left(x_k - I_k v_{oc}(x_k)\frac{T_s}{E_{crit}}\right)\right)^2}{\sigma_\omega^2}}$$

$$= c_1 + \frac{1}{2}\frac{\left(x_{k+1}-\left(x_k - I_k v_{oc}(x_k)\frac{T_s}{E_{crit}}\right)\right)^2}{\sigma_\omega^2}$$

$$-\frac{\partial \log p(x_{k+1}|x_k)}{\partial x_k} = \frac{1}{\sigma_\omega^2}(x_{k+1} - x_k + I_k v_{oc}(x_k)\frac{T_s}{E_{crit}})\left(-1 + I_k \frac{dv_{oc}(x_k)}{dx_k}\frac{T_s}{E_{crit}}\right)$$

$$-\frac{\partial \log p(x_{k+1}|x_k)}{\partial x_{k+1}} = \frac{1}{\sigma_\omega^2}\left(x_{k+1} - x_k + I_k v_{oc}(x_k)\frac{T_s}{E_{crit}}\right)$$

$$\Rightarrow -\frac{\partial^2 \log p(x_{k+1}|x_k)}{\partial {x_k}^2} = \frac{1}{\sigma_\omega^2}\left(\left(-1 + I_k \frac{dv_{oc}(x_k)}{dx_k}\frac{T_s}{E_{crit}}\right)^2 \cdots + \left(x_{k+1} - x_k + I_k v_{oc}(x_k)\frac{T_s}{E_{crit}}\right) I_k \frac{d^2 v_{oc}(x_k)}{d{x_k}^2}\frac{T_s}{E_{crit}}\right)$$

$$\Rightarrow -\frac{\partial^2 \log p(x_{k+1}|x_k)}{\partial x_k \partial x_{k+1}} = \frac{1}{\sigma_\omega^2}\left(-1 + I_k \frac{dv_{oc}(x_k)}{dx_k}\frac{T_s}{E_{crit}}\right)$$

$$\Rightarrow -\frac{\partial^2 \log p(x_{k+1}|x_k)}{\partial {x_{k+1}}^2} = \frac{1}{\sigma_\omega^2}$$

Appendix D. Impact of design hyper- parameter on prognostic algorithm performance

Table D.2
Dissimilarity between predicted state MSE and MCP-BCRLB curves (ℓ^1 distance) for $N_\theta = 500$. Candidates that were discarded in Step 3 are marked with a × symbol. Candidates associated with minimum distances are marked with a √ symbol.

$\|\text{MSE}_{h_{\theta,i}} - \text{MCP-BCRLB}\|_1$	$h_{\theta,1}$	$h_{\theta,2}$	$h_{\theta,3}$	$h_{\theta,4}$	$h_{\theta,5}$
$k_p = 2000$ [s]	7.7933	6.3497	5.1402	4.3521√	4.2905×
$k_p = 4000$ [s]	4.6074	4.6585	4.0858	3.0111√	2.7673×
$k_p = 6000$ [s]	2.1356	2.0592	1.9863	1.6549√	1.7349×

Table D.3
Dissimilarity between predicted state MSE and MCP-BCRLB curves (ℓ^1 distance) for $N_\theta = 1000$. Candidates that were discarded in Step 3 are marked with a × symbol. Candidates associated with minimum distances are marked with a √ symbol.

$\|\text{MSE}_{h_{\theta,i}} - \text{MCP-BCRLB}\|_1$	$h_{\theta,1}$	$h_{\theta,2}$	$h_{\theta,3}$	$h_{\theta,4}$	$h_{\theta,5}$
$k_p = 2000$ [s]	7.0399	6.3170	5.0063	4.7501√	4.1201×
$k_p = 4000$ [s]	4.5221	4.1722	3.5164	2.8865√	2.6738×
$k_p = 6000$ [s]	2.6213	2.1310	2.0551√	1.7470×	1.7253×

Table D.4
Dissimilarity between predicted state MSE and MCP-BCRLB curves (ℓ^1 distance) for $N_\theta = 5000$. Candidates that were discarded in Step 3 are marked with a × symbol. Candidates associated with minimum distances are marked with a √ symbol.

$\|\text{MSE}_{h_{\theta,i}} - \text{MCP-BCRLB}\|_1$	$h_{\theta,1}$	$h_{\theta,2}$	$h_{\theta,3}$	$h_{\theta,4}$	$h_{\theta,5}$
$k_p = 2000$ [s]	7.0030	6.3168	5.3319	4.3004√	4.2133×
$k_p = 4000$ [s]	4.6211	4.4223	3.6820	3.2294√	2.6773×
$k_p = 6000$ [s]	2.5407	2.2773	1.9471√	1.6789×	1.6890×

Table D.5
Dissimilarity between predicted state MSE and MCP-BCRLB curves (ℓ^1 distance) for $N_\theta = 10,000$. Candidates that were discarded in Step 3 are marked with a $\times$ symbol. Candidates associated with minimum distances are marked with a $\surd$ symbol.

$\|\text{MSE}_{h_{\theta,i}} - \text{MCP-BCRLB}\|_1$	$h_{\theta,1}$	$h_{\theta,2}$	$h_{\theta,3}$	$h_{\theta,4}$	$h_{\theta,5}$
$k_p = 2000$ [s]	7.1497	6.1347	5.2790	4.4606$\surd$	4.1559$\times$
$k_p = 4000$ [s]	4.8909	4.1680	3.6592	2.8972$\surd$	2.7126$\times$
$k_p = 6000$ [s]	2.4718	2.0924	1.8365$\surd$	1.6653$\times$	1.6822$\times$

References

[1] D. Liu, V. Luo, Y. Peng, Uncertainty processing in prognostics and health management: an overview, 2012 IEEE Conference on Prognostics and System Health Management (PHM) (2012).
[2] A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.
[3] D.E. Acuña, M.E. Orchard, Particle-filtering-based failure prognosis via sigma-points: application to lithium-ion battery state-of-charge monitoring, Mech. Syst. Signal Process. 85 (2017) 827–848.
[4] M. Orchard, G. Vachtsevanos, A particle- filtering approach for on-line fault diagnosis and failure prognosis, Trans. Inst. Meas. Control 31 (2009) 221–246.
[5] D.A. Pola, H.F. Navarrete, M.E. Orchard, R.S. Rabié, M.A. Cerda, B.E. Olivares, J.F. Silva, P.A. Espinoza, A. Pérez, Particle- filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles, IEEE Trans. Reliab. 64 (2) (2015) 710–721.
[6] S. Engel, B. Gilmartin, K. Bongort, A. Hess, Prognostics, the real issues involved with predicting life remaining, Proc. 2000 IEEE Aerospace Conf. 6 (2000) 457–469.
[7] D.J.C. MacKay, Introduction to Monte Carlo methods, in: Learning in Graphical Models, 1998.
[8] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, G. Vachtsevanos, Advances in uncertainty representation and management for particle filtering applied to prognostics, Proc. Int. Conf. Prognostics and Health Manag. (2008) 1–6.
[9] K. Goebel, A. Saxena, S. Saha, B. Saha, J. Celaya, Prognostic performance metrics, in: Data Mining in Systems Health Management: Detection, Diagnostics, and Prognostics, Chapman & Hall/CRC Press, 2011, pp. 149–171.
[10] C.R. Rao, Information and the accuracy attainable in the estimation of statistical parameters, Bull. Calcutta Math. Soc. 37 (1945) 81–89.
[11] H. Cramér, Mathematical Methods of Statistics, 1st ed., Princeton Univ. Press, Princeton, NJ, 1946.
[12] H.L. Van Trees, Detection, Estimation and Modulation Theory, vol. 1, Wiley, New York, 1968.
[13] C. Fritsche, E. Ozkan, L. Svensson, F. Gustafsson, A fresh look at Bayesian Cramér-Rao bounds for discrete-time nonlinear filtering, 17th International Conference on Information Fusion – FUSION (2014).
[14] P. Tichavský, C.H. Muravchik, A. Nehorai, Posterior Cramér-Rao bounds for discrete-time nonlinear filtering, IEEE Trans. Signal Process. 46 (5) (1998) 1386–1396.
[15] L. Zuo, R. Niu, P. Varshney, Conditional posterior Cramér-Rao lower bounds for nonlinear sequential Bayesian estimation, IEEE Trans. Signal Process. 59 (2011).
[16] D.E. Acuña, M.E. Orchard, Prognostic algorithms design based on predictive Bayesian Cramér-Rao lower bounds, in: 20th IFAC World Congress, July 9th–14th, Tolouse, France, 2017.
[17] M. Šimandl, J. Královec, P. Tichavský, Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time nonlinear dynamic systems, Automatica 37 (2001) 1703–1716.
[18] B.E. Olivares, M.A. Cerda, M.E. Orchard, J.F. Silva, Particle-filtering-based prognosis framework for energy storage devices with a statistical characterization of state-of-health regeneration phenomena, IEEE Trans. Instrum. Meas. 62 (2) (2013) 364–376.
[19] C. Burgos, M.E. Orchard, M. Kazerani, R. Cárdenas, D. Sáez, Particle-filtering-based estimation of maximum available power state in lithium-ion batteries, Appl. Energy 161 (2016) 349–363.
[20] T.K. Moon, W.C. Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice-Hall/Prentice-Hall International (UK), Upper Saddle River, NJ/London, 2000.