Mechanical Systems and Signal Processing 135 (2020) 106421

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

A method for the reduction of the computational cost associated with the implementation of particle-filter-based failure prognostic algorithms

Heraldo Rozas [a,*], Francisco Jaramillo [a], Aramis Perez [a,c], Diego Jimenez [a], Marcos E. Orchard [a], Kamal Medjaher [b]

[a] *Department of Electrical Engineering, Faculty of Physical and Mathematical Sciences, Universidad de Chile, Santiago, Chile*
[b] *Production Engineering Laboratory (LGP), INPT-ENIT, 47 Av. d'Azereix, 65000 Tarbes, France*
[c] *School of Electrical Engineering, University of Costa Rica, San Jose, Costa Rica*

ARTICLE INFO

Article history:
Received 30 January 2019
Received in revised form 8 July 2019
Accepted 4 October 2019
Available online 17 October 2019

Keywords:
Prognostic algorithms
Time-of-Failure probability distribution
Online performance assessment

ABSTRACT

Failure prognostic algorithms require to reduce the computational burden associated with their implementation to ensure real-time performance in embedded systems. In this regard, this paper presents a method that allows to significantly reduce this computational cost in the case of particle-filter-based prognostic algorithms, which is based on a time-variant prognostic update rate. In this proposed scheme, the performance of the prognostic algorithm within short-term prediction horizons is continuously compared with respect to the outcome of Bayesian state estimators. Only if the discrepancy between prior and posterior knowledge is greater than a given threshold, it is suggested to execute the prognostic algorithm once again and update Time-of-Failure estimates. In addition, a novel metric to evaluate the performance of any prognostic algorithm in real-time is hereby presented. The proposed actualization scheme is implemented, tested, and validated in two case studies related to the problem of State-of-Charge (SOC) prognostics. The obtained results show that the proposed strategy allows to significantly reduce the computational cost while keeping the standards in terms of algorithm efficacy.

1. Introduction

System components and industrial equipment undergo natural degradation processes as a direct consequence of their use in daily operation. As these degradation processes may eventually lead to the occurrence of catastrophic events and loss of operational continuity, it has become necessary to prioritize the implementation of State-of-Health (SoH) monitoring architectures and Prognostics and Health Management (PHM) systems [1]. Furthermore, PHM systems have been lately considered as key enabling technologies for the implementation of decision-making schemes on future operating load profiles based the information of current condition of the process (i.e., the outcome of Fault Diagnostic modules) and the characterization of the future evolution of the system SoH (i.e., the outcome of Failure Prognostic modules).

In this regard, it is important to remark that fault diagnostic modules typically perform 3 main tasks: (i) to detect a fault that is affecting the performance of the process, (ii) to isolate and determine in which component is the fault actually located, and (iii) to estimate how severe the damage is, using for this purpose an adequately defined Health Index (HI). Failure

* Corresponding author.
E-mail address: heraldo.rozas@ing.uchile.cl (H. Rozas).

https://doi.org/10.1016/j.ymssp.2019.106421

prognostic modules, on the other hand, are primarily focused on using degradation models to characterize the future evolution of the fault and evaluate the probability of failure at any given time. Failure prognostic algorithms allow to evaluate the reliability of a system in it is actual life cycle conditions, predicting the time at which a system or a component will no longer perform its intended function, thus giving users the opportunity to mitigate system level risks while extending its useful life [1].

Particle-filtering-based Prognostics (PFP) algorithms have been widely applied to solve the problem of real-time characterization of the Time-of-Failure (ToF) probability density function (PDF) [2]. Indeed, these methods are currently considered the state-of-the-art for model-based prognostic by many researchers within the PHM community [3], and have been employed in diverse applications, including railway tracks [4], rotating machines [5], batteries [6–8], among others [9–11]. Although PFP algorithms have been widely studied, the literature still does not offer a formal analysis on the determination of the most appropriate prognostic update rate to be used in order to decrease the associated computational costs without significantly affecting the algorithm efficacy. In this regard, it is important to emphasize that the aim of our work is focused on proposing a novel method to reduce the computational effort associated with the implementation of real-time particle-filter-based prognostic algorithms, this being a differentiating factor compared to other efforts currently reported in the literature [12–14]. In other words, our objective is to introduce a methodology capable of determining the need of an update in Time-of-Failure estimates, considering for this purposes an assessment on the capabilities of the prognostic module to predict the degradation of the faulty system in a short-term prediction horizon.

To contextualize this problem, let us assume an online implementation where we can acquire real-time measurements, such as in the case of online End-of-Discharge (EoD) time prognostic problem for Li-ion batteries [7]. In this case, it has been demonstrated that state estimates can be sequentially updated using any of the filtering algorithms available in the literature (e.g., Bayesian processors). However, a reasonable question would be to determine if it is actually necessary to run the EoD time prognostic algorithms every time a new observation is acquired (as conventional implementations of PFP algorithm suggest [15,16]).

Indeed, one way to solve the aforementioned dilemma is to assume that accurate prognostic results require to incorporate even the smallest innovation perceived during the filtering stage, since *"failure prognostic results should be continuously updated using the latest available measurement information. New information should be incorporated to improve the knowledge about the system, including distribution of the model parameters, system performance, and future loading conditions. This information is critical for the accurate prognostics"* [17]. Nevertheless, in practical situations it would result complicated to implement this procedure due to the high computational cost. Moreover, in real-time embedded systems, with limited computational resources, this decision would be counterproductive [18,19].

Therefore, it may be interesting to investigate whether there is an optimal moment, or even an optimal update rate, that needs to be considered in terms of the implementation of failure prognostic algorithms. To answer this question, this article proposes a novel time-varying prognostic rate actualization scheme that depends on the performance of the latest characterization of the evolution of the system in time. In this scheme, the performance of the prognostic algorithm within short-term horizons is continuously compared with the outcome of Bayesian processors that perform the task of state estimation. If the difference between prior and posterior knowledge is greater than a given threshold, and only then, it is suggested to update the prognostic result by executing the failure prognostic module once more. The underlying idea there in is "to run prognostic if, and only if, is necessary". In addition, and as a byproduct of this research effort, this article also proposes a novel metric to evaluate the performance of any prognostic algorithm in real-time.

Therefore, the main objectives of our work can be summarized as follows:

- To provide an efficient and practical procedure to define the moment when it is required to update the latest results provided by the prognostic algorithm.
- To provide a metric for online assessment of failure prognostic algorithms.

The proposed method is implemented, tested, and validated in two case studies related to the problem of State-of-Charge (SOC) prognostics. Obtained results show that the proposed strategy allows to significantly reduce the computational cost while keeping the standards in terms of algorithm efficacy.

The article structure is as follows. Section 2 focuses on providing theoretical background on particle filtering, particle-filtering-based prognostic algorithms, and online prognostic performance metrics. Section 3 presents the proposed methodology for Real-time Assessment of Probability-based Failure Prognostic Algorithms. Section 4 shows the implementation of the proposed scheme in two case studies related to the problem of SOC prognostics in Li-ion batteries (a.k.a. the EoD prognostic problem). Finally, Section 5 presents the main conclusions of this research work.

2. Theoretical background

2.1. The particle filter

In engineering applications, Bayesian filtering is a mathematical framework for recursively estimating the evolving posterior distribution of the target state of a dynamical system [20]. In practice, these dynamical systems are typically non-linear and non-Gaussian, which entails that the solution of the Bayesian filtering problem (or also called optimal solution) cannot be computed analytically [21].

A widely known and powerful algorithm used to obtain a sub-optimal solution for the Bayesian filtering problem is the Particle Filter (PF). This kind of filter, based on Monte Carlo simulation, aims to represent the posterior distribution of the target state by a set of random samples (called particles) with associated weights [22].

More formally, PF is a kind of algorithm proposed to obtain samples sequentially from a target state probability distribution $\pi_k(x_{0:k})$ with the purpose of generating a set of $N \gg 1$ weighted particles $\{w_k^{(i)}, x_{0:k}^{(i)}\}_{i=1,\ldots,N}$, $w_k^{(i)} > 0$, $\forall k \geqslant 1$, satisfying Eq. (1) [23]

$$\sum_{i=1}^{N} w_k^{(i)} \varphi_k \left(x_{0:k}^{(i)}\right) \underset{N\to\infty}{\to} \int \varphi_k(x_{0:k})\pi_k(x_{0:k})d_{x_{0:k}}, \tag{1}$$

where φ_k is any π_k-integrable function. And, for the case of the Bayesian filtering problem, the target distribution is chosen as $\pi_k(x_{0:k}) = p(x_{0:k} \mid y_{1:k})$, the posterior PDF of the state vector conditional to noisy observations $y_{1:k}$ [22].

Let us assume that at time $k-1$ a set of N paths (particles) $\{x_{0:k-1}^{(i)}\}_{i=1,\ldots,N}$ are available. Moreover, these paths are distributed according to $q_{k-1}(x_{0:k-1})$ (also called the importance density at time $k-1$). Then, PFs are used to efficiently obtain a set of N new paths $\{\tilde{x}_{0:k}^{(i)}\}_{i=1,\ldots,N}$ distributed approximately according to $\pi_k(\tilde{x}_{0:k})$ [23].

For this, the current paths $x_{0:k-1}^{(i)}$ are extended using the kernel $q_k(\tilde{x}_{0:k} \mid x_{0:k-1}) = \delta(\tilde{x}_{0:k-1} - x_{0:k-1}) \cdot q_k(\tilde{x}_k \mid x_{0:k-1})$, with $\tilde{x}_{0:k} = (x_{0:k-1}, \tilde{x}_k)$. And including concepts of importance sampling, the posterior PDF ($\pi_k(x_{0:k})$) can be approximated through the following empirical distribution (Eq. (2)) [23]

$$\tilde{\pi}_k^N(x_{0:k}) = \sum_{i=1}^{N} w_{0:k}^{(i)} \delta(x_{0:k} - \tilde{x}_{0:k}^{(i)}) \tag{2}$$

where $w_{0:k}^{(i)} \propto w_{0:k}(\tilde{x}_{0:k}^{(i)})$ and $\sum_{i=1}^{N} w_{0:k}^{(i)} = 1$.

The most basic PF implementation, the sequential importance sampling (SIS) [22], assumes that the importance density function equal to the *a priori* state transition PDF, ie, $q_k(\tilde{x}_{0:k} \mid x_{0:k-1}) = p(\tilde{x}_k \mid x_{k-1})$. In this way, the particle weights $w_{0:k}^{(i)}$ are computed using the likelihood of the new observations. The efficiency of the procedure improves as the variance of the particle weights is minimized [23].

2.2. Failure prognostic based on particle filters

In PHM, a failure prognostic algorithm is understood as a method that is able to characterize future uncertainty sources that may affect the evolution of degraded systems in time and generate long-term predictions of the state vector (fault indicator), everything with the purpose of estimating the remaining useful life (RUL) of a failing component/subsystem [2]. Failure prognostic algorithms typically use information provided by Bayesian filtering modules to determine a reasonable initial condition for those long-term predictions, since Bayesian processors allow to fuse prior knowledge on the degradation model structure with real-time measurements that are acquired once the fault has been detected. However, prognostic algorithms need to characterize the most likely sequence of state vector probability density functions for future time instants in absence of additional observations and, thus, their design and implementation entails a series of challenges that go beyond the typical definition of a Bayesian filtering scheme (e.g., sequential Monte Carlo methods).

In this context, particle-filtering-based prognostic algorithms are a family of methods widely accepted real-time solution to the failure prognostic problem within the PHM community. PF-based prognostic algorithms assume a nonlinear model to describe the evolution of the fault indicator (i.e., the state vector) and non-Gaussian sources of uncertainty [7]; offering a family versatile tools that can be applied to a variety of problems, including crack propagation, bearing degradation, and battery monitoring systems.

A particle-filtering-based prognostic algorithm is basically a method for future uncertainty characterization that uses sequential Monte Carlo to obtain a state *posterior* PDF during the *filtering* stage. PFP algorithms use the particle population as a probability mass function for the state, and provide a way to propagate particles in time to represent the evolution of this initial uncertainty in a long-term prediction problem.

The math behind the implementation of failure prognostic algorithms has been properly described in [24]. In this regard, let us imagine a system that can incur into a catastrophic failure condition only once. Similarly to the experiment of tossing coin, at each time instant k the system may continue operating or not. We denote $\mathcal{H}_k$ as the event of *being in a faulty, although operative*, condition at time k, whereas $\mathcal{F}_k$ denotes the event of *undergoing a catastrophic failure* at time k.

Given the above, it is possible to approximate the true probability of failure at the k-th time instant, $\mathcal{P}(\mathcal{F}_k)$. Then, the probability of catastrophic failure can be computed as:

$$\mathcal{P}(\mathcal{F}_k) = \mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1})\mathcal{P}(\mathcal{H}_{k_p:k-1}), \tag{3}$$

where k_p is the time at which the prognostic algorithm is executed.

Now, since $\mathcal{P}(\mathcal{H}_{k_p:k-1})$ is the probability of having the system still operative at the $(k-1)$-th time instant -which means a finite intersection of events-, and using the properties of conditional probabilities, we can approximate:

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

and where

$$\mathcal{P}(\mathcal{H}_j | \mathcal{H}_{k_p:j-1}) = 1 - \mathcal{P}(\mathcal{F}_j | \mathcal{H}_{k_p:j-1}), \quad \forall j > k_p.$$

Because of the mutual exclusion among operative and failed conditions, it follows that:

$$\mathcal{P}(\mathcal{F}_k) \cong \mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1}) \prod_{j=k_p+1}^{k-1} \left(1 - \mathcal{P}(\mathcal{F}_j | \mathcal{H}_{k_p:j-1})\right). \tag{4}$$

As it can be observed in Eq. (4), any failure probability measure is fully determined by understanding the meaning of $\mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1})$, for all k. It is important to note that in this mathematical notation we assume and omit, on purpose, the conditional on the set of measurements $y_{1:k_p}$ in all expressions.

Considering all of the above, particle-filtering-based prognostic methods can be defined as a class of algorithms that allow to properly approximate $\mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1})$ by a collection of weighted particles $\{W_k^{(i)}, x_k^{(i)}\}_{i=1,\dots,N}$, such that:

$$\mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1}) = \sum_{i=1}^{N_p} W_k^{(i)} \mathcal{P}(\mathcal{F}_k | \mathcal{H}_{k_p:k-1}, X_k = x_k^{(i)}). \tag{5}$$

A key element in the implementation of PF-based prognostic methods is the procedure to compute the weights $W_k^{(i)}, i = 1, \dots, N$ in Eq. (5). A detailed description of at least three methods that can be used for this purpose is found in [2]. The most simple (and common) method is to use the weights of the particles that characterize the posterior state PDF (computed during the filtering stage) and keep them constant in the generation of long-term predictions.

As stated above, the execution of PFPs algorithms incorporates the previous filtering stage as the initial condition for the prognostic stage. Therefore, for each new measurement acquired in real-time, a new filtering step is applied; also allowing to update the prognostic stage (Fig. 1). This procedure may improve the characterization of the failure time PDF (since better estimates of the posterior state PDF are available and also because of a shorter prognostic horizon). However, this procedure requires high computational resources to be executed.

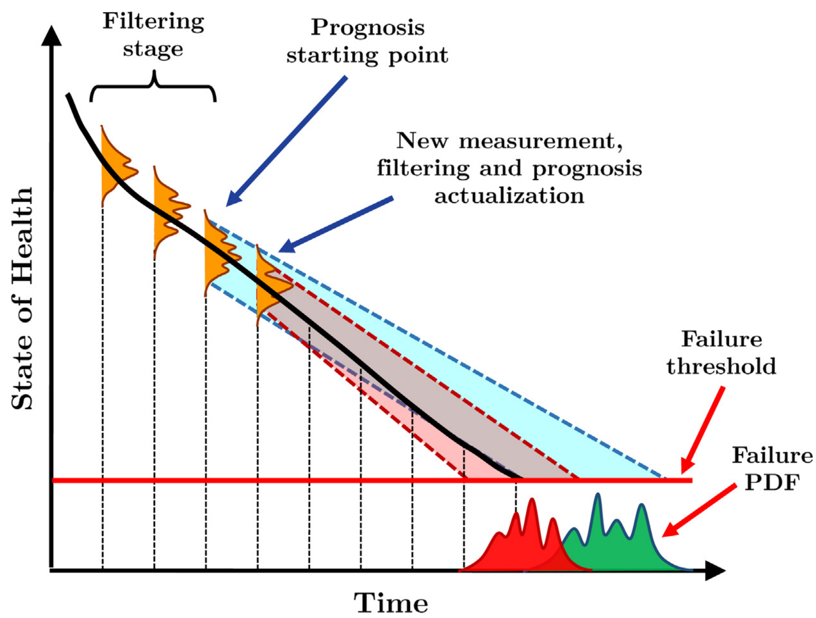

Fig. 1. Graphic illustration of a PFP algorithm execution. PFP algorithms use the current *posterior* PDF estimate as initial condition. As new measurement are acquired, it is possible to update the *posterior* state PDF and, if desired, the outcome of prognostic modules.

2.3. Prognostic algorithm performance metrics

In literature, researchers have clustered prognostic algorithm performance metrics using different criteria. This study separates these metrics based on the ability to perform offline vs. online performance assessment. In the first group (offline performance assessment metrics), the "ground truth" evolution of the health index in time is known beforehand and thus, it is intensively used to measure the effectiveness of any given prognostic method [25]. A variety of classical metrics, for example $\alpha - \gamma$ performance, accuracy, convergence, among others [26–28,25], are widely used by the PHM community for this purpose. Nevertheless, in several industrial applications, it may be extremely difficult to acquire data sets characterizing the entire degradation process until a catastrophic failure and, thus, these metrics will not be applicable to the problem that motivates this research effort [29].

The second group, online prognostic performance metrics aim to provide a solution to the issue described above. For online assessment purposes, the "ground truth" failure time is unknown and, consequently, the definition of metrics is extremely challenging. PHM researchers in the past have declared this problem for a number of years [26,25,28] and nevertheless, to the best of our knowledge, developments on this area have been scarce. In fact, so far, only two recent publications have made advances in this regard (see [29,30]).

In [29], the authors proposed an online prognostic performance metric based on a comparison between the prognostic execution and the current degradation trajectory. For comparison purposes, they define a sliding time window where estimates of the degradation process has been computed. However, within this sliding window, authors compare the prognostic module outcome against degradation process estimates by using classical offline prognostic performance metrics, a choice that is counterproductive.

On the other hand, in [30], prognostic algorithm performance is compared against the outcome of simulations from a dynamic model with equivalent process noise characterization (basically, a Monte Carlo simulation with a restricted number of realizations). Although theoretically speaking the concept behind this proposal is correct, it becomes difficult to ensure adequate real-time assessment of prognostic algorithms when using the latter approach; mainly due to the number of simulations that are required to properly characterize future uncertainty in dynamic nonlinear systems.

3. Determining the need of an update in time-of-failure estimates

This research effort aims to develop a novel time-varying prognostic rate actualization scheme capable of providing information regarding the need of an update on ToF estimates, considering for this purpose performance assessments of the latest result generated by failure prognostic algorithms. To do this, it is also necessary to propose an adequate method to assess the quality of probability-based failure prognostic algorithms in real-time.

The proposed solution should achieve its goal while simultaneously meeting the following requirements:

- Offer an adequate balance between prognostic performance and algorithm computational cost.
- Must use less computational resources than the traditional schemes where prognostic results are computed every time a measurement is acquired.

3.1. Degradation model structure

The proposed scheme is applicable to model-based prognostic algorithms. Thus, it assumes that the evolution in time of the HI of the monitored system can be well-described in terms of a state-space characterization. Thereby, for all practical purposes, the degradation model structure is as follows:

State Transition equation:

$$x_t = f(x_{t-1}, u_{t-1}, w_{t-1}) \tag{6}$$

Observation Equation:

$$y_t = g(x_t, v_t) \tag{7}$$

where $f(\cdot)$ is the state transition function, $g(\cdot)$ is the state observation function, u_t is the input of the system; while w_t and v_t are the process and observation noise, respectively. Functions $f(\cdot)$ and $g(\cdot)$ are non-linear, while w_t and v_t are non-Gaussian noises. As this model structure enables the implementation of particle-filtering-based diagnostic and prognostic schemes (see SubSection 2.2), this research effort will assume that uncertainty sources will always be characterized by a set of $N \gg 1$ weighted particles $\{w_k^{(i)}, x_{0:k}^{(i)}\}_{i=1,\dots,N}, \; w_k^{(i)} > 0, \quad \forall k \geqslant 1$.

3.2. Proposed solution for the determination of optimal update times in failure prognostic algorithms

Before presenting the proposed scheme for failure prognostic algorithms update requests, it is important to generate an intuition about the developed strategy. In this regard, it becomes relevant to understand which are the main factors that may affect the quality of prognostic results. Assuming that the system degradation model is well-characterized (see SubSection 3.1), we have 2 factors that have a direct influence on the quality of ToF estimates:

- Initial conditions for the state vector in the state-space model [7].
- Characterization of the uncertainty associated with future system inputs (i.e., future operating profiles) [25].

Taking this into account, one question arises: is it actually possible improve our knowledge on any of these factors as more system measurements are acquired? Actually, yes. Indeed, Bayesian processors (such as particle-filtering schemes) allow to continuously improve the characterization of initial conditions for long-term predictions in prognostic algorithms. Conversely, if the evolution of the system in the short-term closely follows predicted trajectories for the state vector, which means that it is not expected to observe significant discrepancies between the posterior state PDF and the predicted (prior) state PDF, then initial conditions for long-term predictions should be similar to those that were anticipated. As similar initial conditions for long-term predictions will reflect into similar ToF estimates, it would not be advisable to run a prognostic update (and spend the significant computational resources related to this task) unless there is evidence that the future operating profile is significantly different from what it was assumed. Moreover, an analogous analysis can be made for the case when the characterization of the uncertainty associated with future system inputs is unaltered.

Inspired by this intuition, the following solution to the problem of determining the need of an update in ToF estimates is now proposed. At each sample time, a short-term predicted state PDF is compared against the filtering state PDF using an *ad hoc* online prognostic performance metric. This metric will basically measure the difference between these state probability distributions. If the trajectories associated with the latest prognostic results available are close to the outcome of the filtering process, then the advisory system will recommend to avoid an update on ToF estimates. Otherwise, the prognostic results will be updated using the current filtering state estimate as initial condition for long-term predictions. Fig. 2 depicts an illustration of the proposed scheme, where the quality of the latest available prognostic result is assessed in terms of its capability to describe the current state of the system by an online prognostic performance metric. If the outcome of the assessment, Θ, is lower than a predefined threshold, $\mathcal{M}$, then an update of ToF estimates is suggested.

Note that in this scheme, the *ad hoc* online prognostic performance metric plays an important role. This metric will be described in detail in the following subsection.

3.3. Proposed online prognostic performance metric

The main problem regarding the implementation of online performance assessment procedures is that the actual trajectory of the state vector is unknown. Nevertheless, it is important to note that reliable characterizations of the evolution of the damage progression should perform well both in the long-term (naturally required to estimate the ToF) and in the short-term. Inspired by this fact, a novel online prognostic performance metric is defined by measuring the discrepancy between short-term predicted state PDFs and the outcomes of the filtering stage once sufficient measurements are acquired. This metric follows some of the concepts described in [29], regarding the fact that it compares prognostic results with PDF estimates, but it differs in the manner in which the metric is actually computed.

Indeed, considering that this research effort assumes the implementation of PF-based algorithms to compute the posterior PDF of the state vector, and that uncertainty is characterized by a set of weighted particles, we propose the following performance metric:

1. Compute the probability interval of λ percent for the posterior state PDF around its expectation. Let α be the lower bound of this interval and β its upper bound.
2. Consider a short-term prediction for the state PDF that was computed L time instants ago (with L negligible with respect to the prognostic horizon). Compute the probability mass (Θ_k) of this predicted PDF that falls within the probability interval $[\alpha, \beta]$. This can be simply done by adding the weights of the particles that fall within the interval.

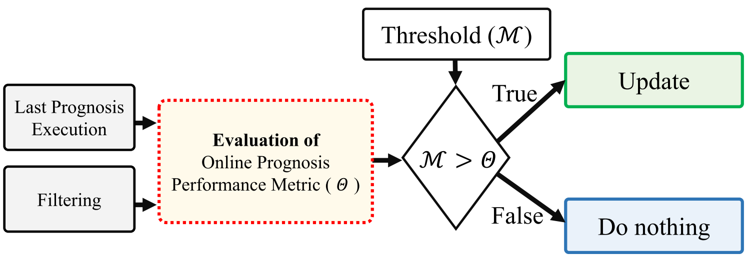

Fig. 2. Graphic abstract for the proposed method for the determination of optimal update times in failure prognostic algorithms.

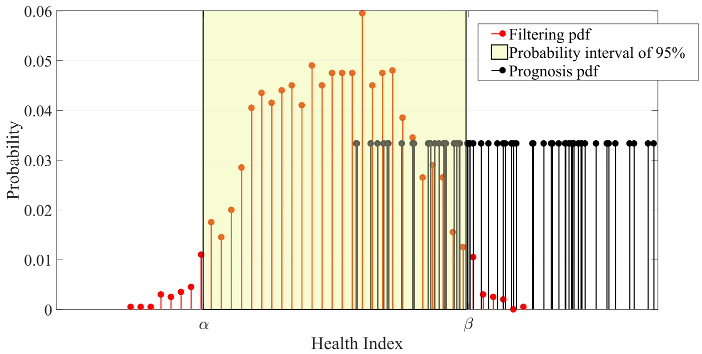

Fig. 3. Graphical representation of the proposed online prognostic performance metric, for $\lambda = 0.95$. Please note that all particles in the PF-based prognostic algorithm have identical weights.

Please note that a similar metric has been defined in [25] for offline prognostic performance assessment. This research redefines the latter metric for online applications using only partial knowledge on the degradation process.

A graphical representation of the concept behind this new prognostic metric can be found in Fig. 3.

The values of α and β (bounds of the probability interval of λ percent for the posterior state PDF around its expectation) can be computed as follows:

$$\mathcal{P}(x_k \leqslant \alpha) = \frac{(1-\lambda)}{2} \tag{8}$$

$$\mathcal{P}(x_k \geqslant \beta) = \frac{(1-\lambda)}{2} \tag{9}$$

where $\mathcal{P}(x_k)$ is the posterior PDF estimate.

Note that the parameter λ is defined by the user. It is important to note that the state vector prediction represents prior knowledge about the future condition of the system, which is more uncertain than the outcome of the filtering stage (posterior knowledge). As result, the proposed metric should take values in the interval $[0, \lambda]$, where λ is the best case and 0 is the worst situation, which corresponds to a case where prior knowledge differs completely from posterior knowledge in the support defined by the probability interval. Two main reasons justify the selection of this specific performance metric:

- The metric takes into account the similarity between state PDF estimates obtained from filtering stage and short-term predictions computed by the prognostic algorithm under analysis. If the metric value is close to 0.95, both PDFs will have similar supports for their respective distributions. If ToF estimates are updated in this situation, no significant changes should be expected; given that the initial condition for long-term predictions will be almost equivalent to the one that was anticipated for the current time instant.
- The metric has a low computational cost. This is important because one of the most important objectives of this work is to propose an efficient way to solve the problem.

Perhaps the best way to illustrate the advantages of the proposed scheme is to analyze its performance in terms of the efficiency and efficacy of obtained prognostic results in actual case studies. Particularly, this research effort has chosen the problem of EoD time prognostic in Li-ion battery cells for this purpose, motivated by the fact that this problem is still a matter of ongoing research within the Prognostic and Health Management community [24,31–33,8].

4. Case of study: end-of-discharge time prognostic in lithium-ion battery cells

This section focuses on the implementation and validation of the proposed prognostic scheme in the context of battery EoD time prognostic problems (a.k.a SOC prognostic). In particular, two case studies are now presented. The first one corresponds to an experiment conducted under carefully controlled conditions, while in the second the SoC prognostic problem is studied in the context of an application for an electric bike (real-life usage). These case studies offer the opportunity to incorporate additional sources of uncertainty and perform more complete sensitivity analyses: (1) impact of initial conditions on the quality of prognostic results, (2) characterization of unknown future exogenous inputs via probabilistic methods, (3) the opportunity to compare our proposed approach with respect to the conventional method in terms of computational effort, and (4) sensibility analysis respect to parameters λ and $\mathcal{M}$.

4.1. EoD Time prognostic in Li-ion battery

EoD Time prognostic is a widely studied problem within the PHM community [24,31–33,8]. In this case, the aim is to evaluate the amount of energy that still remains on a Li-ion battery cell (filtering problem) and prognosticate the moment at which that energy will be depleted (failure prognostic problem). For these purposes, filtering and prognostic modules 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 battery cell typically is modeled as a Thévenin equivalent circuit, where the voltage source is a function of the state, specifically the SOC. 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. However, the state-space model proposed in [7] is used in this research effort instead, allowing to characterize the nonlinear behavior present in this curve. The resulting state-space model is as follows:

State Transition Equation:

$$x_1(k+1) = x_1(k) + \omega_1(k) \tag{10}$$

$$\begin{aligned} x_2(k+1) = x_2(k) - \big(v_l + (v_o - v_l)e^{\gamma(x_2(k)-1)} + \ldots\ldots + \alpha v_l(x_2(k)-1) + (1-\alpha)v_l(e^{-\beta} - e^{-\beta\sqrt{x_2(k)}}) + \ldots\ldots \\ - i(k)x_1(k) + \eta(k)\big)i(k)\Delta t E_{crit}^{-1} + \omega_2(k) \end{aligned} \tag{11}$$

Observation equation:

$$v(k) = v_l + (v_o - v_l)e^{\gamma(x_2(k)-1)} + \alpha(k)v_l(x_2(k)-1) + \ldots\ldots + (1-\alpha(k))v_l(e^{-\beta} - e^{-\beta\sqrt{x_2(k)}}) - i(k)x_1(k) + \eta(k) \tag{12}$$

where x_1 and x_2 are the internal resistance of the battery cell and the SOC, respectively. In addition, the parameters $\alpha, \beta, v_0, v_l, \gamma$ and E_c are estimated off-line following the procedure described in [7].

The aforementioned space-state model for the battery discharge process allows to implement particle-filtering-based prognostic algorithms. In particular, without loss of generality, this article assumes the utilization of the approach suggested in [7].

4.2. Case study 1: battery cell discharged in laboratory

The battery used in the experiment was a Panasonic Li-ion cell (specifically type CGR18650CH). The parameters that characterize this cell are shown in Table 1. They are determined by an offline experimental test that follows the procedure described in [7].

A particle-filtering estimation and prognostic scheme was implemented to keep track of the SOC and predict the moment in which the battery cell will be completely discharged. In this case, the algorithm uses 200 particles for this purpose. Failure prognostic algorithms also require to characterize the future evolution of exogenous inputs to the state-space model (future operating profiles). In this case study, future inputs for the system assume a constant discharge current profile, whose value is computed as the average of the latest 30 measurements of the actual battery discharge current.

4.2.1. Algorithm performance assessment: obtained results

The implementation of filtering-prognostic scheme based on PF was performed on a desktop computer, with an Intel i5 3 (Ghz) processor and 8(GB) of RAM. Note that in the implementation, the parameter $\mathcal{M}$ (threshold for prognostic update decision) is set as $\mathcal{M} = 0.8$, while λ is set as $\lambda = 0.95$. As a consequence, an update of the ToF estimate will be suggested every time that the sum of the weights associated with a L-step short-term prediction of the state PDF falls within the probability interval $[\alpha, \beta]$, is lower than 0.8.

The results of the proposed methodology for prognostic update rate are shown both in Fig. 4 and Table 2. The window time in this case study corresponds to 854[s]. Considering a conventional update rate for prognostic results, we would need to run the prognostic algorithm 854 times. The implementation of the proposed methodology to determine an appropriate update rate resulted in solely 6 executions of the prognostic routine.

The time of prognostic executions and time lapse during which each prognostic result was valid are found in the second and third column of Table 2, respectively. Each time that prognostic is computed the metric is maximum (for instance at time

Table 1
Estimated battery cell parameters. Experimental results were obtained by applying the procedure described in [7].

Parameter	Value	Unit
E_c	24464	J
v_l	3.997	V
v_0	4.14	V
α	0.15	
β	17	
γ	10.5	

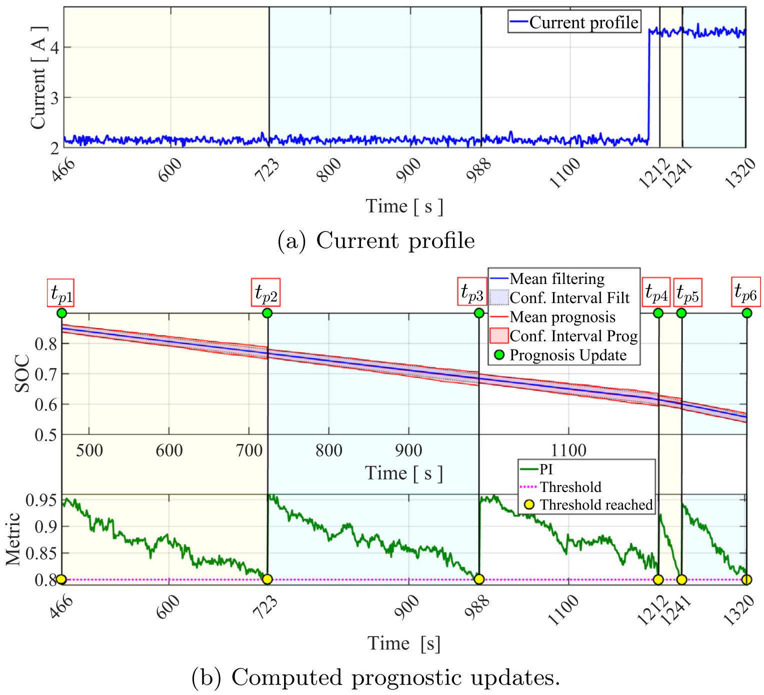

Fig. 4. Results of implementation of proposed methodology for prognostic update rate.

Table 2
Time lapse during which successive prognostic results were valid.

Prognostic algorithm run number	Time of prognostic algorithm execution [s]	Time lapse during which results were valid [s]
1	466	257
2	723	265
3	988	224
4	1212	29
5	1241	82

723[s], see Fig. 4b), then it starts to decrease due to the prognostic uncertainty increment. The metric decay rate is highly depending on the prognostic performance; it can be better analyzed in terms of the time lapse during which successive prognostic results were valid. For prognostic execution i, it was computed as $t_{pi+1} - t_{pi}$. On the one hand, it is interesting to emphasize the fact that the first 3 executions of the prognostic algorithm are valid during time intervals of similar length (in fact, each of them lasted more than 200[s]). On the other hand, the last 2 executions of the prognostic routine were valid for a considerably less amount of time. The latter is caused by a change in the operating conditions of the battery cell. Indeed, the characterization of the future discharge current profile is computed as the average of the latest 30 measured values of this exogenous variable, and at time 1200[s] occurs a sudden and significant change in the battery discharge current (see Fig. 4a). At that time, the assumptions considered by the prognostic module (in terms of the future battery discharge current profile) were inexact and, therefore, the performance of the prognostic algorithm degraded and the proposed approach for determination of update rates in the execution of prognostic algorithms requested an actualization of EoD estimates more often.

This experiment helps to illustrate the main advantage of the proposed prognostic scheme, when compared to other conventional implementations. Particularly, it is possible to observe how the proposed methodology evaluates the discrepancy between prior and posterior knowledge, requesting an update of the prognostic result only if necessary, with the consequent savings in computational cost. Nevertheless, as this experiment is conducted under carefully controlled conditions, it cannot be used to conclude on the usefulness of the proposed methodology in other circumstances. For this reason, we have decided to perform a second experiment, where the SOC prognostic problem is studied in the context of an application for an electric bike (real-life usage).

4.3. Case study 2: battery EoD prognostics in an electric bicycle

This case study studies the performance of the proposed algorithm in the context of a real-time SOC monitoring application for the battery of an electric bicycle (real-life usage). The battery is instrumented with voltage and current sensors at its terminals, and data is acquired during one complete discharge cycle under normal usage conditions. Similarly to the procedure described in the former case study, the first step for the implementation of the proposed methodology is to estimate the parameters that characterize the Lithium-Ion battery. These parameter values are summarized in Table 3.

As opposed to the first case study that was presented in this article, in this experiment the discharge profile cannot be characterized via a constant value due to the inherent variability associated with the usage of the bicycle. Consequently, the characterization of future exogenous inputs is a more challenging problem. Nevertheless, this objective can be achieved using a Markov Chain stochastic model and following the steps described in [7]. The resulting stochastic model allows to characterize future inputs for the system through a set of realizations obtained from the Markov Chain; see Fig. 5.

4.3.1. Algorithm performance assessment: obtained results

The implementation of the proposed filtering and prognostic scheme used in this case the same default values for parameters ($\mathcal{M} = 0.8$ and $\lambda = 0.95$) than the first application example.

In this test, the initial battery SOC is 0.8 (i.e., 80%). However, and with the purpose of analyzing the impact of erroneous initial conditions, the PF-based estimator assumes a biased SOC initial condition equal to 0.7. The proposed prognostic update methodology is compared with a "conventional" approach that executes the prognostic algorithm each time a new measurement is acquired. It is important to remark that both update schemes utilize the same PF-based prognostic algorithm to predict the battery EoD, being the update rate the only difference.

Fig. 6 shows obtained results for this experiment during both filtering and prognostic stages. First, in Fig. 6(a), it is possible to observe that the estimator is able to correct for the erroneous initial condition and converge to ground truth SOC value in approximately 100 s. However, there is also evidence indicating that this error in SOC estimates is detrimental for the quality of prognostic results computed at early stages. Indeed, as the SOC posterior estimate is biased, so is the result of the prognostic algorithm. As a consequence, and particularly during the first 100 s of operation, the proposed prognostic methodology recommends to perform updates every 15 s in average; see Fig. 6(b). This update rate diminishes noticeably afterwards, limiting the execution of prognostic algorithms only after 139 s in average.

This phenomenon can be explained in terms of the differences found between long term predictions and filtered estimates for the battery SOC during early stages, where the likelihood of real-time measurements provide critical information about the accuracy of prior estimates. In contrast, once SOC estimates has converged, the innovation associated with each newly acquired measurements is less significant (as long as the state equation provides a reasonable model for the system), given the fact that the battery discharge process is slow compared to the sampling period. The proposed methodology is able to capture this information, which explains the reduction in the recommended prognostic update rate after the 100th second.

Table 3
Electric parameters for the battery of electric bicycle. Estimates were obtained following the procedure described in [7].

Parameter	Value	Unit
E_c	1279900	J
v_l	33.481	V
v_0	41.405	V
α	−0.005	
β	11.505	
γ	1.553	

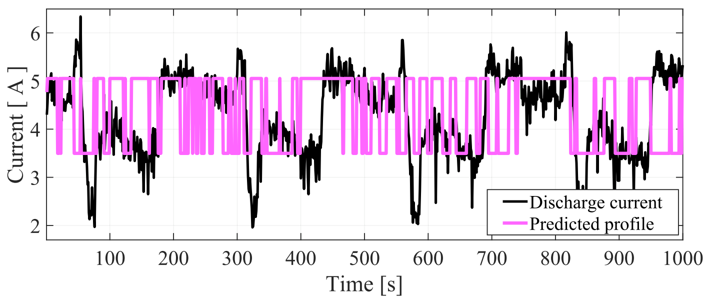

Fig. 5. Characterization of future discharge current using a realization of a Markov Chain model.

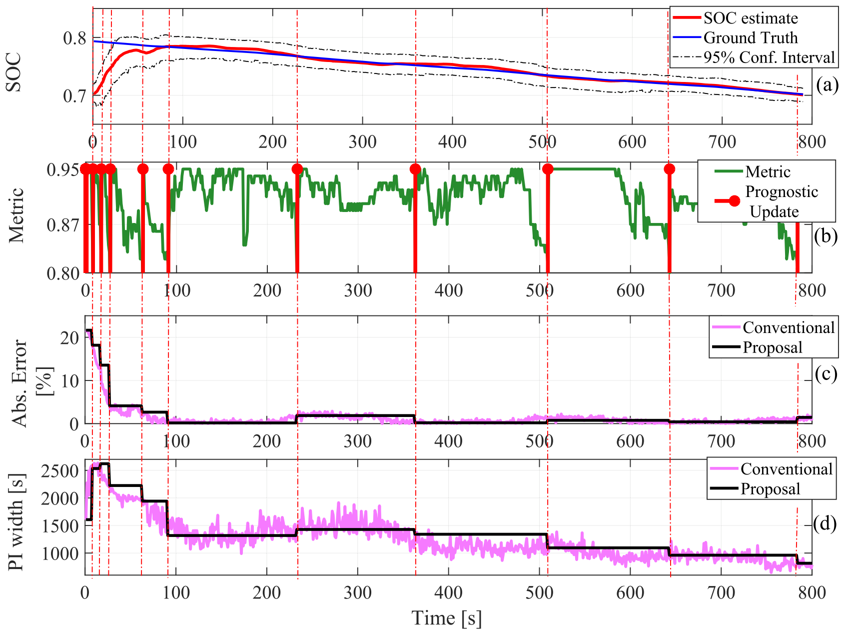

Fig. 6. Comparison between proposed and conventional prognostic update schemes.

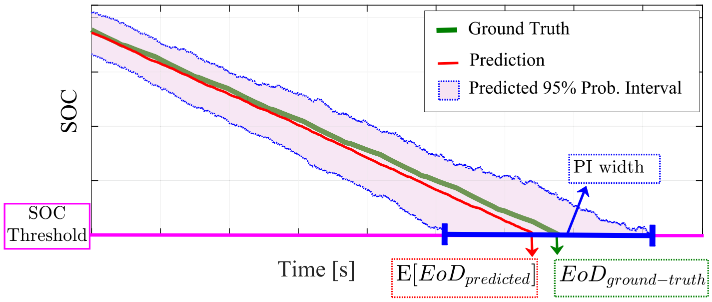

Fig. 7. Illustration of figures of merit used to measure the effectiveness of prognostic results.

A quick assessment of the computational effort associated with the implementation of the proposed methodology indicates that, during 800 s of operation, EoD prognostic results had to be updated only 11 times (compared to 800 updates that would be needed when using conventional real-time approaches). In this regard, it is evident that our proposal has a lower computational cost than conventional update schemes.

Now, the question that naturally follows from the previous analysis is: what can be said in terms of the efficacy of the obtained prognostic results? To answer this question, two figures of merit will be used; allowing us to measure the predictive capability in terms of the End-of-Discharge time (EoD). These figures of merit are defined as follows:

1. Absolute Error (Abs. Error):

$$\frac{|\mathrm{E}[EoD_{predicted}] - EoD_{ground-truth}| \cdot 100\%}{EoD_{ground-truth}},$$

   where $\mathrm{E}[EoD_{predicted}]$ is expectation for the time when the predicted SOC reaches the critical discharge threshold, whereas $EoD_{ground-truth}$ is the "ground truth" value associated with the moment in which this event happens. These quantities are illustrated in Fig. 7.
2. Probability Interval Width (PI width): Corresponds to the width of 95% probability interval for the predicted EoD (see Fig. 7).

The intuition behind these figures of merit, which are always non-negative, is illustrated in Fig. 7. Prognostic results are deemed to be effective if both metrics are small (close to zero).

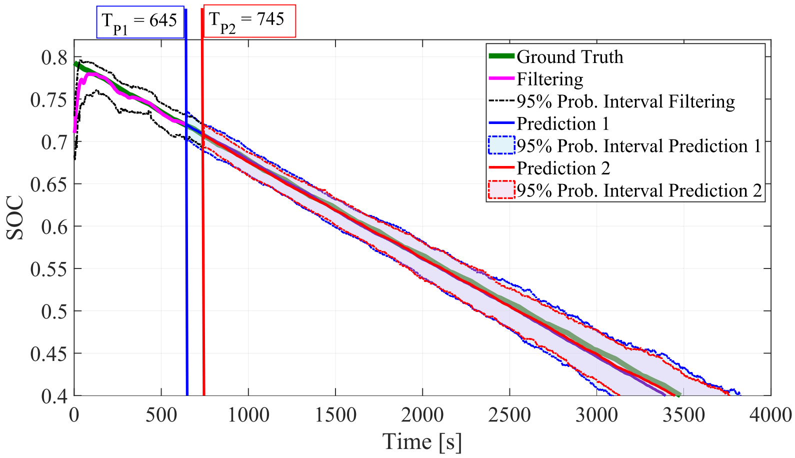

Fig. 8. Comparison between two prognostic results computed at different times.

On the one hand, if conventional approaches to real-time failure prognostics are used, the EoD prediction is updated every time a new measurement (voltage and discharge current) is acquired. Consequently, at each sample time, both metrics (Abs. Error and PI width) could be evaluated. On the other hand, the proposed approach recommends to execute prognostic algorithms only when the threshold update is reached. In this regard, at each sample time, the performance of the proposed scheme is measured solely in terms of the latest result available.

Fig. 6(c) and (d) shows obtained results in terms of "Abs. Error" and "PI width", respectively. It is noteworthy that both schemes exhibit poor performances at early stages of the experiment, being this fact a consequence of the utilization of erroneous initial conditions during the filtering stage. Nevertheless, as the estimates converge to the ground truth SOC, both figures of merit decrease significantly. In addition, it is possible to confirm that the uncertainty related to prognostic results decreases as the prediction horizon does, since the "PI width" decreases noticeably as the experiment reaches its ground truth EoD.

An analysis based on "Abs. Error" (see Fig. 6(c)) indicates that both schemes have similar performance in average. Please note that conventional schemes generate results that sometimes are better than the proposed methodology, but the opposite is also true. Also, almost identical behavior is observed in term of "PI width" (see Fig. 6(d)), although the conventional approach offers slightly better performance in average.

In summary, both schemes have a similar performance (effectiveness) in terms of the figures of merit included in this study. The only difference being that the conventional implementation of prognostic algorithms obtains a slight reduction in the width of the probability interval at the very end of the experiment (caused by the inclusion of samples during the last seconds of operation). Nevertheless, this difference is negligible in terms of decision-making strategies that could be implemented in the usage of electric bicycle. To emphasize the latter statement, please refer to the situation depicted in Fig. 8, where two outcomes from the prognostic algorithm are considered: "Prediction 1", computed at time 645[*s*], and "Prediction 2", computed at time 745[*s*]. Even though both prognostic results were computed at different time instants, the associated performance is almost identical.

At this point, and considering the aforementioned analysis, it is possible to conclude that even in cases where the exogenous input of the system is unknown and characterized in terms of a stochastic process, the proposed methodology for prognostic updates (1) uses significantly less computational resources than the conventional approach, and (2) exhibits similar performance in terms of its effectiveness. Thus, it is possible to state that the proposed scheme is, indeed, more efficient.

4.3.2. Sensitivity analysis

The proposed methodology uses two parameters (λ and $\mathcal{M}$) to define the recommended prognostic update policy. In this regard, it is important to understand the role that these parameters pay in terms of the efficiency of the scheme. This issue is addressed in the sensitivity analysis that follows.

The sensitivity analysis aim at studying the effect of parameters λ and $\mathcal{M}$ in terms of the number of prognostic updates that the method requires within a fixed time window. This analysis is performed, in this case, using data from the experiment of SOC monitoring in the electric bicycle. The time window used in the analysis is 944 s and obtained results are shown in Table 4.

From Table 4, one the one hand, for a fixed value of parameter λ, it is observed that when the threshold $\mathcal{M}$ decreases, the prognostic update rate also diminishes. This result is expected, since when we consider smaller thresholds for the probability of being inside the interval associated with the posterior distribution, it basically implies that the prognostic update scheme is being less strict with respect to the minimum acceptable precision and accuracy of predictions for the state vector. On the other hand, for a fixed threshold $\mathcal{M}$, we observe that when the probability interval λ decreases, the prognostic update rate

Table 4
Number of prognostic updates required in a 944 s time window.

λ	$\mathcal{M}$			
	0,90	0,80	0,70	0,60
0,99	8	7	5	4
0,95	37	10	7	5
0,90	256	21	6	5
0,85	699	32	13	7
0,80	944	119	25	8

increases. The latter result is also expected, since a smaller probability interval in the posterior distribution implies that the prognostic update scheme is being more strict. Moreover, we note that if parameters are set with the most strict combination (λ decreases while $\mathcal{M}$ increases), the proposed prognostic update scheme results equivalent to conventional prognostic update rate method.

5. Conclusion

This article presents a novel time-varying prognostic rate actualization scheme that depends on the performance of the latest characterization of the evolution of the system in time. The proposed actualization scheme is implemented, tested, and validated in two case studies related to the problem of battery SOC prognostics.

Obtained results show that the proposed strategy allows to significantly reduce the computational cost while keeping the standards in terms of algorithm efficacy; proving the value behind research efforts that address the problem of appropriate updates rates in prognostic algorithm implementations. In this regard, it is highly recommended to incorporate the proposed procedure in embedded applications with limited computational resources, and also as a tool to provide guidelines in terms of prognostic algorithm design.

In addition, the proposed online performance metric demonstrates its merit as a tool to assess the predictive capabilities associated with a specific prognostic algorithm implementation; being its main advantage the low computational cost and the straightforward interpretation. Future research work will be oriented towards the utilization of this metric in the formulation of an optimal prognostic update rate problem.

Acknowledgements

This work has been supported by FONDECYT Chile Grant Nr. 1170044, CONICYT REDES 170031, and the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, CONICYT. The work of Heraldo Rozas was supported by CONICYT-PFCHA/MagisterNacional/2018-22180232. The work of Francisco Jaramillo was supported by CONICYT-PCHA/Doctorado Nacional/2014-21140201. The work of Aramis Perez was supported by the University of Costa Rica, and CONICYT-PCHA/Doctorado Nacional/2015-21150121.

References

[1] N.-H. Kim, D. An, J.-H. Choi, Prognostics and Health Management of Engineering Systems, Springer International Publishing, Switzerland, 2017, https://doi.org/10.1007/978-3-319-44742-1.
[2] M. Orchard, G. Vachtsevanos, A particle-filtering approach for on-line fault diagnosis and failure prognosis, Trans. Inst. Meas. Control. 31 (2009) 221–246, https://doi.org/10.1177/0142331208092026.
[3] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, N. Zerhouni, Particle filter-based prognostics: review, discussion and perspectives, Mech. Syst. Signal Process. 72–73 (2016) 2–31, https://doi.org/10.1016/j.ymssp.2015.11.008.
[4] M. Mishra, J. Odelius, A. Thaduri, A. Nissen, M. Rantatalo, Particle filter-based prognostic approach for railway track geometry, Mech. Syst. Signal Process. 96 (2017) 226–238, https://doi.org/10.1016/j.ymssp.2017.04.010.
[5] W. Caesarendra, G. Niu, B.-S. Yang, Machine condition prognosis based on sequential monte carlo method, Expert Syst. Appl. 37 (2010) 2412–2420, https://doi.org/10.1016/j.eswa.2009.07.014.
[6] B.E. Olivares, M.A.C. Munoz, 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, Instrum. Meas. IEEE Trans. 62 (2013) 364–376, https://doi.org/10.1109/TIM.2012.2215142.
[7] 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 (2015) 710–720, https://doi.org/10.1109/TR.2014.2385069.
[8] C. Sbarufatti, M. Corbetta, M. Giglio, F. Cadini, Adaptive prognosis of lithium-ion batteries based on the combination of particle filters and radial basis function neural networks, J. Power Sour. 344 (2017) 128–140, https://doi.org/10.1016/j.jpowsour.2017.01.105.
[9] J. Fan, K.-C. Yung, M. Pecht, Predicting long-term lumen maintenance life of led light sources using a particle filter-based prognostic approach, Expert Syst. Appl. 42 (2015) 2411–2420, https://doi.org/10.1016/j.eswa.2014.10.021.
[10] M. Rigamonti, P. Baraldi, E. Zio, D. Astigarraga, A. Galarza, Particle filter-based prognostics for an electrolytic capacitor working in variable operating conditions, IEEE Trans. Power Electron. 31 (2016) 1567–1575, https://doi.org/10.1109/TPEL.2015.2418198.
[11] C.P. Ley, M.E. Orchard, Chi-squared smoothed adaptive particle-filtering based prognosis, Mech. Syst. Signal Process. 82 (2017) 148–165, https://doi.org/10.1016/j.ymssp.2016.05.015.
[12] E.N. Chatzi, A.W. Smyth, Particle filter scheme with mutation for the estimation of time-invariant parameters in structural health monitoring applications, Struct. Control Heal. Monit. 20 (2013) 1081–1095, https://doi.org/10.1002/stc.1520.

[13] B.G. Sileshi, J. Oliver, C. Ferrer, Accelerating particle filter on FPGA, in: 2016 IEEE Comput. Soc. Annu. Symp. VLSI, 2016, pp. 591–594, https://doi.org/10.1109/ISVLSI.2016.66.
[14] A. Olivier, A.W. Smyth, Particle filtering and marginalization for parameter identification in structural systems, Struct. Control Heal. Monit. 24 (3) (2017), https://doi.org/10.1002/stc.1874 e1874.
[15] C. Chen, D. Brown, C. Sconyers, G. Vachtsevanos, B. Zhang, M.E. Orchard, A.NET framework for an integrated fault diagnosis and failure prognosis architecture, in: 2010 IEEE Autotestcon, 2010, pp. 1–6, https://doi.org/10.1109/AUTEST.2010.5613626.
[16] G. Bartram, S. Mahadevan, Probabilistic prognosis with dynamic bayesian networks, Int. J. Progn. Heal. Manage. 6 (2015) 1–23.
[17] T. Peng, J. He, Y. Xiang, Y. Liu, A. Saxena, J. Celaya, K. Goebel, Probabilistic fatigue damage prognosis of lap joint using bayesian updating, J. Intell. Mater. Syst. Struct. 26 (2015) 965–979, https://doi.org/10.1177/1045389X14538328.
[18] W. Yan, B. Zhang, W. Dou, D. Liu, Y. Peng, Low-cost adaptive lebesgue sampling particle filtering approach for real-time li-ion battery diagnosis and prognosis, IEEE Trans. Autom. Sci. Eng. 14 (2017) 1601–1611, https://doi.org/10.1109/TASE.2017.2666202.
[19] W. Yan, B. Zhang, X. Wang, W. Dou, J. Wang, Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries, IEEE Trans. Ind. Electron. 63 (2016) 1804–1812, https://doi.org/10.1109/TIE.2015.2494529.
[20] J. Ko, D. Fox, GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models, Auton. Robots 27 (2009) 75–90, https://doi.org/10.1007/s10514-009-9119-x.
[21] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter: particle filters for tracking applications, Artech House (2003).
[22] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Trans. Signal Process. 50 (2002) 174–188, https://doi.org/10.1109/78.978374.
[23] C. Andrieu, A. Doucet, E. Punskaya, Sequential Monte Carlo methods for optimal filtering, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, New York, 2001, pp. 79–95, https://doi.org/10.1007/978-1-4757-3437-9_4.
[24] 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, https://doi.org/10.1016/j.ymssp.2016.08.029.
[25] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, Metrics for offline evaluation of prognostic performance, Int. J. Progn. Heal. Manage. 1 (2010) 1–20.
[26] K. Goebel, A. Saxena, S. Saha, B. Saha, J. Celaya, Prognostic performance metrics, in: A.N. Srivastava, J. Han (Eds.), Machine Learning and Knowledge Discovery for Engineering Systems Health Management, CRC Press, Boca Raton, FL, USA, 2011, pp. 149–171.
[27] A. Saxena, J. Celaya, E. Balaban, K. Goebel, B. Saha, S. Saha, M. Schwabacher, Metrics for evaluating performance of prognostic techniques, in: 2008 Int. Conf. Progn. Heal. Manag., 2008, pp. 1–17, https://doi.org/10.1109/PHM.2008.4711436.
[28] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, On applying the prognostic performance metrics, in: Annu. Conf. Progn. Heal. Manag. Soc. 2009, 2009, pp. 1–13.
[29] Y. Hu, P. Baraldi, F. Di Maio, E. Zio, Online performance assessment method for a model-based prognostic approach, IEEE Trans. Reliab. 65 (2016) 718–735, https://doi.org/10.1109/TR.2015.2500681.
[30] Y. Wang, C. Gogu, N.H. Kim, R.T. Haftka, N. Binaud, C. Bes, Noise-dependent ranking of prognostics algorithms based on discrepancy without true damage information, Reliab. Eng. Syst. Saf. 184 (2019) 86–100, https://doi.org/10.1016/j.ress.2017.09.021.
[31] S.S. Mansouri, P. Karvelis, G. Georgoulas, G. Nikolakopoulos, Remaining useful battery life prediction for UAVs based on machine learning, IFAC-PapersOnLine 50 (2017) 4727–4732, https://doi.org/10.1016/j.ifacol.2017.08.863.
[32] M.E. Orchard, M.A. Cerda, B.E. Olivares, J.F. Silva, Sequential monte carlo methods for discharge time prognosis in lithium-ion batteries, Int. J. Progn. Heal. Manage. 3 (2012) 90–101.
[33] C. Tampier, A. Pérez, F. Jaramillo, V. Quintero, M.E. Orchard, J.F. Silva, Lithium-ion battery end-of-discharge time estimation and prognosis based on bayesian algorithms and outer feedback correction loops: a comparative analysis, in: Annu. Conf. Progn. Heal. Manag. Soc. 2015, 2015, pp. 1–14.