Comparison of different models of future operating condition in Particle-Filter-based Prognostic Algorithms [⋆]

Heraldo Rozas [*] Ferhat Tamssaouet [**] Francisco Jaramillo [*]
Khanh T.P. Nguyen [**] Kamal Medjaher [**] Marcos Orchard [*]

[*] *Universidad de Chile, Tupper 2007, Santiago, Chile*
(e-mail: {heraldo.rozas, francisco.jaramillo, morchard}@ing.uchile.cl)
[**] *Laboratoire Génie de Production, Université de Toulouse,*
INP-ENIT, Tarbes, France
(e-mail: {ferhat.tamssaouet, thi-phuong-khanh.nguyen,
kamal.medjaher}@enit.fr)

Abstract: In literature, a major part of the prognostic studies considers the mission profile as a static parameter when evaluating the system Remaining Useful Life (RUL). However, in practice, the way in which a system operates significantly impacts the future evolution of its degradation. Therefore, this paper aims at evaluating the impact associated with the utilization of three different methods to characterize future operating conditions within the implementation of probability-based prognostic algorithms, namely Long-short term memory (LSTM), Markov Chain and Constant (or time-invariant) usage. These three methods are compared together in terms of both prognostic accuracy and essential update times when investigating the Time-of-Discharge (ToD) of an electric bicycle Lithium-Ion (Li-Ion) battery.

Keywords: Failure prognostic; Mission profile; Markov Chain; LSTM network; Li-Ion battery.

NOMENCLATURE

Abs. Error	Absolute Error
BF	Bayesian Filtering
EoD	End-of-Discharge
Li-Ion	Lithium-Ion
LSTM	Long-short term memory
PDF	Probability Density Function
PF	Particle Filter
PFP	Particle-filtering-based Prognostics
PHM	Prognostics and Health Management
PI Width	Probability Interval Width
RNNs	Recurrent Neural Networks
RUL	Remaining Useful Life
SIS	Sequential Importance Sampling
SOC	State-of-Charge
SoH	State-of-Health
ToD	Time-of-Discharge
ToF	Time-of-Failure
UAVs	Unmanned Aerial Vehicles

1. INTRODUCTION

Due to the increasing requirements of reliability and safety of industrial systems, it is essential to implement the suitable as well as, efficient and accurate Prognostics and Health Management (PHM) solutions for a reliable State-of-Health (SoH) and Remaining Useful Life (RUL) estimation. These solutions must be able to represent the dynamics of the defined health index, along with a proper characterization of all relevant sources of uncertainty, such as modeling, future inputs, and prediction method (Celaya et al., 2012; Sankararaman, 2015). Prognostic algorithms play a major role due to their capability of estimating the components RUL, whereby they yield valuable information for both decision-makers (Kordestani et al., 2019).

One key challenge in the design of prognostic algorithms is their computational burden. Note that real-time execution is desirable, even mandatory, for some systems, such as mission re-planning in Unmanned Aerial Vehicles (UAVs) and terrain robots. Unfortunately, these systems often have limited computational resources on-board. Therefore, for such applications, the main goal of prognostic designers is to develop algorithms capable of computing effective predictions within a reasonable computational time. Endeavoring real-time prognostic algorithms, authors have explored a great variety of approaches. One of the most promising attempts is presented in Orchard and Vachtsevanos (2009), where authors proposed an algorithm based on Particle Filter (PF) for computing online prognostic, which estimates the failure time Probability Density Function (PDF) based on a state space derived from physical

[⋆] This work has been supported by FONDECYT Chile Grant Nr. 1170044, ANID REDES 170031, and the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, ANID. The work of Heraldo Rozas was supported by ANID-PFCHA/Magister Nacional/2018-22180232. The work of Francisco Jaramillo was supported by ANID-PCHA/ Doctorado Nacional/2014-21140201.

laws. This proposal tries to reduce the computational cost of Monte Carlo's simulation by using PF and regularization. This approach is considered as the state-of-the-art in model-based prognostic by many researchers within the PHM community (Jouin et al., 2016; Tamssaouet et al., 2019; Pola et al., 2015), having applications in many engineering domains. Most of these reserach efforts show promising results in terms of prediction capability, but the computational burden is not treated as a priority, and moreover, it is even not analyzed. Another interesting approach to deal with real-time prognostics was introduced in Yan et al. (2015). This approach computes prognostics through Lebesgue-sampling-based procedure, wherein prediction steps are discretized in the state space, but not time space. As a result, significant computation efforts can be saved; indeed, up-and-coming results are presented for batteries prognostics in Yan et al. (2015). Nevertheless, this approach demands a dynamic model in Lebesgue space whose description could be highly challenging.

On the other hand, a completely different insight is explored in (Rozas et al., 2020), in which the authors reduce substantially the computational burden of real-world implementations of Particle-filtering-based Prognostics (PFP) algorithms by the utilization of a time-varying prognostics update rate. To be more specific, under conventional PFP implementations, the prognostics algorithm has to be executed at each sampling time, whereas, under the approach developed in Rozas et al. (2020), the prognostic outcomes are updated over a time-varying rate that directly depends on the performance of the latest prognostics execution. The underlying idea can be described as follows. If the last prognostics execution has a good performance, then its corresponding prognostic results will be valid for a long period, and hence no updates are required. Conversely, if the last prognostics execution presents a poor performance, then a new prognostics execution, so-called prognostics update, will be shortly required. To evaluate the performance of the PFP executions, a novel real-time metric was indeed proposed. The methodology proposed in (Rozas et al., 2020) is committed to reduce the computational burden of real-world implementations of PFP algorithms, and was implemented, tested, and validated successfully in two case studies related to the problem of battery State-of-Charge (SOC) prognostics. Despite its promising outcomes, this proposal lacks a discussion about the relevance of the characterization of future operating conditions into the prognostics performance. Note that if we have a most robust characterization of the future operating profile, it may derive better prognostics results that require less prognostics updates, and thus lead to an even more important computational burden reduction. Consequently, in the present work, we will investigate the impact of 3 different models of future operating profile on the prognostic results computed by a PFP algorithm. The objective is, therefore, to complement the proposal developed in (Rozas et al., 2020).

Additionally, we should consider that PFP algorithms have a great diversity of uncertainty sources, which entail a challenging task in the design and implementation of PFP algorithms. Being more specific, we have to deal with four major sources of uncertainty: 1) degradation model structure and parameters, 2) initial condition for the state at the time of prediction (typically solved through filtering techniques such as PF), 3) the prognostic algorithm itself (which is a simplification of Monte Carlo methods), and 4) future operating profiles (Sankararaman, 2015). Most authors have extensively proposed methods and solutions for the first three, but little attention has been paid in terms of the latter. As a consequence, the present work will also contribute to the discussion about the impact of the future operating profile on the uncertainty characterization.

As a summary, and considering the aforementioned gaps in the literature, this article seeks to study the effects of different models of future operating conditions characterization on both prognostic accuracy and computational efficiency. More specifically, with the aim of characterizing operating profile in the SOC prognostic problem using PFP, three models are investigated 1) Long-Short-Term Memory (LSTM), 2) Markov-Chain, and 3) constant (or time-invariant) usage profile. The motivation of studying varied models is the fact that future operating conditions is one of the largest sources of uncertainty for prognostic algorithms (Daigle and Sankararaman, 2013), thus its appropriate modelling may allow performing more efficient prognostics in terms of a more accurate Time-of-Failure (ToF) and RUL estimations (Sankararaman, 2015; Sierra et al., 2019).

The article is structured as follows. Section 2 aims to present a theoretical background of the proposed methodology such as PFP, the LSTM network, and the prognostic update procedure developped in Rozas et al. (2020). Section 3 investigates the case study for which the proposed methodology is applied to handle the uncertainty of future operating conditions and to study the effect of these modeling approaches on the prognostic performance. Section 4 focuses on the analysis of the obtained results. Finally, in Section 5, conclusions are presented.

2. THEORETICAL BACKGROUND

This section aims to present the theoretical background based on which the proposed methodology is developed. In detail, the concepts of Bayesian and Particle filters are recalled in subsection 2.1. Next, subsection 2.2 is dedicated to describe the PF based prognostics approach. The LSTM structure is briefly presented in subsection 2.3. Finally, subsection 2.4 focuses on the prognostic update procedure.

2.1 Bayesian filtering and Particle Filters

Bayesian Filtering (BF) is a methodology for recursively estimating the *posterior* PDF of the state vector $\mathbf{x}_k$ at each time step k given the measurements $\mathbf{z}_{1:k}$ up to time k. Thus, the so-called BF problem consists in computing the PDF $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. In practical applications, the dynamics of $\mathbf{x}_k$ is usually non-linear, time-variant and subject to non-Gaussian uncertainties, implying that the optimal solution of the BF problem cannot be solved analytically (Arulampalam et al., 2002; Särkkä, 2013).

A widely-used method to obtain a sub-optimal solution for the BF problem is the PF. This method, based on Monte Carlo simulation, seeks to represent the *posterior* PDF of $\mathbf{x}_k$, at each time step k, by a set of N_p random samples with associated weights, called *particles.* To accomplish this, the PF sequentially obtain a set of *particles* $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^{N_p}$, $\sum_{i=1}^{N_p} w_k^{(i)} = 1$ from an alternative $q(\cdot)$ PDF, called importance PDF. Therefore, the *posterior* PDF of $\mathbf{x}_k$ is represented by (Arulampalam et al., 2002):

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \sum_{i=1}^{N_p} w_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}), \qquad (1)$$

where weights $w_k^{(i)}$ are updated according to (Sequential Importance Sampling (SIS) (Arulampalam et al., 2002)):

$$w_k^{(i)} = w_{k-1}^{(i)} p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) \qquad (2)$$

2.2 Particle Filtering-based Prognostics

In PHM, the main objective of failure prognostic algorithms is to characterize future operational risk based on long-term predictions for the evolution of a set of fault indicators. The PFP algorithm (Orchard and Vachtsevanos, 2009) is a method that uses sequential Monte Carlo to describe the propagation of uncertainty over time using: 1) a stochastic state-space model of the faulty system, 2) an initial condition for the state at the instant time at which the prognostic algorithm is executed (extracted from the *filtering* stage), and 3) a characterization of the future operating profiles.

The PFP algorithm presented in Orchard and Vachtsevanos (2009) works as follows. Prognostics executed at time k starts with a set of weighted-particles $\{x_k^i, w_k^i)\}_{i=1}^{N_p}$, where N_p is the number of particles, x_k^i is the state of particle i, and w_k^i is the weight of particle i. Then, the particles are propagated step by step according to the stochastic state-space model of the faulty system. For instance, given $\tau \in \{1, ..., n\}$, if we want to predict the state-PDF at time $k+\tau$, we need to propagate the particles from $k+\tau-1$ to $k+\tau$:

$$\hat{p}(x_{k+\tau}|\hat{x}_{k+\tau-1}) \approx \sum_{i=1}^{N_p} w_{k+\tau-1}^i \cdot \hat{p}\left(\hat{x}_{k+\tau}^i|\hat{x}_{k+\tau-1}^i\right) \qquad (3)$$

Note that $\hat{p}\left(\hat{x}_{k+\tau}^i|\hat{x}_{k+\tau-1}^i\right)$ is related to the evolution-equation of the system thus the proper characterization of the futures operating profile plays a critical role.

2.3 Long-short term memory networks

LSTM network is a class of the Recurrent Neural Networks (RNNs), which are widely used to model dependencies in sequential and time-series data. However, RNNs intrinsically have a vanishing gradient problem, disabling them to learn long-term dependencies. The LSTM layer proposed by Hochreiter and Schmidhuber (1997) solves this problem by offering a more complex internal state representation. Each cell of the LSTM has a combination of four layers: a memory and three gates, which provides them the ability to selectively learn, unlearn, or retain information. First, the forget gate uses a sigmoid function to determine which information h_{t-1} from the previous cell state should be forgotten by the current cell memory. Next, the input gate controls the information flow to the memory using a point-wise multiplication operation of sigmoid and tanh functions, respectively. Finally, the output gate selects the inputs x_t and memory information c_t that should be transmitted to the next cell state.

2.4 Prognostics update procedure

As previously mentioned, in real-time implementations of PFP algorithms, we are continuously receiving new measurements, which are used to compute state estimations by running PF at each sampling time. Considering this fact, a reasonable question is: should we also execute the prognostic algorithm at the same rate? This question was addressed in Rozas et al. (2020), where the authors, motivated by reducing computational costs associated with PFP implementations, showed that it is possible to execute prognostic updates at lower rates while keeping the standards in terms of prognostic efficacy. Being more specific, they proposed a strategy that considers time-varying prognostic update rates. Firstly, it compares the available state prediction PDF obtained from the last prognostic execution against the filtering state PDF, estimated by a real-time prognostic performance metric. Then, if the metric evaluation exceeds a user-defined threshold, the prognostic algorithm is executed; otherwise, the latest prognostic execution remains valid.

Additionally, to implement this time-varying prognostic update strategy, the authors introduced a real-time metric to assess prognostic executions. This metric calculates a λ-probability interval bounded by α and β around the expectation of the filtering state; computing the probability of the predicted state in the interval $[\alpha, \beta]$ as follows:

$$\mathcal{M}_k = \sum_{i \in \{n=1,...,N_p | x_{k,pred}^n \in [\alpha,\beta]\}} w_{k,pred}^i \qquad (4)$$

3. CASE STUDY METHODOLOGY

This work aims at comparing different models for the characterization of future uncertainty in operating conditions, and evaluating the impact of these models on prognostic results. This impact will be measured in terms of prognostics update times and prognostics efficacy. In this section, we will present the case study, three models for the characterization of future operating conditions, and the metric used to measure prognostic efficacy.

3.1 Case of study: State-of-Charge prognostic problem in Lithium-Ion batteries

Lithium-Ion (Li-Ion) batteries are becoming increasingly popular, being utilized in diverse applications. The operation of these battery-driven systems strongly depends on the SOC indicator, which represents the percentage of energy available in the battery. Thus, having information about both the current SOC and predictions of its future evolution could be useful, even imperative, to make decisions about the operation of these systems. Particularly, users are interested in prognosticating the End-of-Discharge (EoD) time, which is the moment at which that energy will be depleted (failure prognostic problem). Motivated by the previous point, our case of study is focused on the SOC prognostic for the battery of an electric bicycle during real-life usage.

To accomplish our goal, we apply the methodology presented in Pola et al. (2015), which uses a state-space model to describe the Li-Ion battery voltage over time as a function of (1) the SOC, (2) the battery internal impedance, and (3) the discharge current, that can be considered as the system's operating condition. As a result, we obtain the following state-space model:

State transition equations:

$$x_1(k+1) = x_1(k) + \omega_1(k) \tag{5}$$

$$x_2(k+1) = x_2(k) - v(k) * i(k)\Delta t\, E_{crit}^{-1} + \omega_2(k) \tag{6}$$

Measurement equation:

$$\begin{aligned} v(k) &= v_l + (v_o - v_l)e^{\gamma(x_2(k)-1)} + \alpha v_l(x_2(k)-1) \\ &+ (1-\alpha)v_l(e^{-\beta} - e^{-\beta\sqrt{x_2(k)}}) - i(k) * x_1 + n(k) \end{aligned} \tag{7}$$

where x_1 and x_2 are the internal resistance of the battery and the SOC, respectively. Parameters $\alpha, \beta, v_0, v_l, \gamma$ and E_c are characteristic of each battery, which can be estimated off-line following the procedure described in Pola et al. (2015). Parameters values in the case study presented in this paper are $E_c = 1279900$[J], $v_l = 33.481$ [V], $v_0 = 41.405$ [V], $\alpha = -0.005$, $\beta = 11.505$, and $\gamma = 1.553$.

It is noteworthy to remark that the discharge current is solely available during the filtering stage because we cannot predict which will be the exact future usage. Thus, for SOC prognostic purposes, we have to utilize a model to characterize the evolution of this operating condition. Particularly, this case of study considers the operation of an electric bicycle; thus, the discharge current strongly depends on the type of route.

3.2 Characterizing future operating conditions

To characterize future operating conditions at prognostic stage, three models (LSTM, Markov Chain, and Constant model) are tested. These models are now presented.

LSTM: To train the LSTM using L measurements of the discharge current I (from 1 to t_p), we consider a sliding window of a length l to obtain $L - l + 1$ training sequences. Each window is associated with one target, i.e. the l + 1th measurement. Once the model tuned, it is used to generate long-term predictions of future operating conditions, as shown in Algorithm 1. To achieve this, the last measurement sequence (i.e. from $L - l$ to L-th measured current) is utilized to predict the output $L + 1$, which is added to the sequence with a noise ω. This is repeated recursively until horizon time (T_{max}) is reached.

Markov Chain: The implementation of Markov Chain for the system inputs requires the estimation of transition probabilities, and the definition of maximum and minimum discharge currents. For this purpose, battery discharge current data is segmented by regular time intervals. Each interval contains a fixed number of samples N_w. This segmentation generates m time intervals such that $m = [L/N_w]$, where L is the number of measurements available at the moment. Then, for each j-th interval ($j = 1, \ldots, m$), the minimum ($i_{low}^{(j)} = min\{i'(k)\}$) and maximum ($i_{high}^{(j)} = max\{i'(k)\}$) discharge current values are computed, where $k = 1, \ldots, N_w$, is a time index valid within the j-th window. Next, on each interval, current measurements are quantized into two possible values defined by $i_{low}^{(j)}$ and $i_{high}^{(j)}$. These values define the low-energy, and high-energy consumption states of the Markov Chain that characterizes the j-th interval. Discharge current data satisfying $i' > (i_{low}^{(j)} + i_{high}^{(j)})/2$ are quantized as $i_{high}^{(j)}$. Otherwise, they are quantized as $i_{low}^{(j)}$. For each interval, it is possible to compute transition probabilities p_{ij} between low-energy and high-energy consumption states. These transition probabilities are estimated using maximum likelihood as described in Pola et al. (2015).

Algorithm 1 Future current profile prediction using LSTM

```
inputs: Current previously measured $\{I_1, ..., I_{t_p}\}$
Output: Current predicted $I^{Pred} = \{I_{tp+1}, ..., I_{T_{max}}\}$
$LSTM_{Trained}$ ←Train LSTM with $\{I_1, ..., I_{t_p}\}$
$I = \{I_1, ..., I_{t_p}\}$
for $i = t_p + 1, ..., T_{max}$ do
    $I_i = LSTM_{Trained}(\hat{I} = \{I_{i-l}, ..., I_{i-1}\}) + \omega_i$
return $I^{Pred} = \{I_{t_p+1}, ..., I_{T_{max}}\}$
```

Constant: This model is a simplistic approach, which assumes that the future exogenous inputs can be characterized as a constant value equals to the average of the last thirty measurements.

3.3 Assessing prognostic efficacy

Following the study in Rozas et al. (2020), to evaluate the prognostic efficacy, two figures of merit will be utilized. These metrics allow assessing the prognostic capability in terms of the EoD. These figures of merit are defined as:

(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.

(2) Probability Interval Width (PI Width): It corresponds to the width of 95% probability interval for the predicted EoD.

Prognostic results are considered to be effective if both metrics are close to zero.

4. RESULTS

The results are summarized in Figure 1. On the one hand, Figure 1(a) shows the outcomes associated with the filtering stage. One can find that SOC is being estimated appropriately because SOC filtering is always close to Ground Truth, more specifically, the mean error between both quantities is less than 2 %. On the other hand, Figures 1(b),(c),(d), and (e) depict prognostic-related outcomes regarding both (1) the time-varying prognostic update strategy implemented with three different models to characterize future operating conditions and (2) "Conventional" implementation, which considers that the prognostic algorithm has to be executed at each sampling time.

Figure 1(b) shows the evaluation of the real-time prognostic performance metric (further details in Section 3.3) over time. The trends of these three models are almost identical, in the sense that the metric assessment is high after each prognostic update, and then, as time progresses, it decreases until reach the threshold, which was defined as 0.85 in this implementation. However, the Constant model exhibits a higher prognostic update rate that is evidenced by observing Figure 1(c), which summarizes the cumulative number of prognostic updates computed at each time. This phenomenon could be explained due to the oversimplistic approach considered in the Constant model, which is not able to properly characterize actual future inputs. As a result, the states predicted are not consistent with the filtering states and, consequently, more prognostic updates have to be computed.

Figures 1(d) and (e) exhibit the results associated with the evaluation of the two figures of merit previously defined (see Section 3.3), namely, Abs. Error and PI Width. Firstly, it is relevant to observe in Figure 1(d) that both Markov Chain and LSTM models present similar promising results in terms of the Abs. Error, which is, most of the time, less than 2%. But, even more important, the difference between these two models, that use the same time-varying prognostic update strategy, and Conventional implementation is not significant. Thus, they allow us to considerably reduce the computational costs by executing prognostic at a time-varying rate while keeping acceptable prognostic efficacy in terms of Abs. Error. Indeed, in the case of study, we should have needed 2500 prognostic executions under the conventional prognostic update approach. However, using time-varying updates, we compute less than 10 updates. On the other hand, the Constant model presents a higher Abs. Error. In fact, it was saturated to 4% in the first 1000 s in order to show all of the quantities on the same scale, but real values exceed by 20 %. In fact, the poor performance of the Constant model is due to oversimplified assumptions.

Observing Figure 1(d), it presents results regarding PI Width. As to the previous figure, both LSTM and Markov

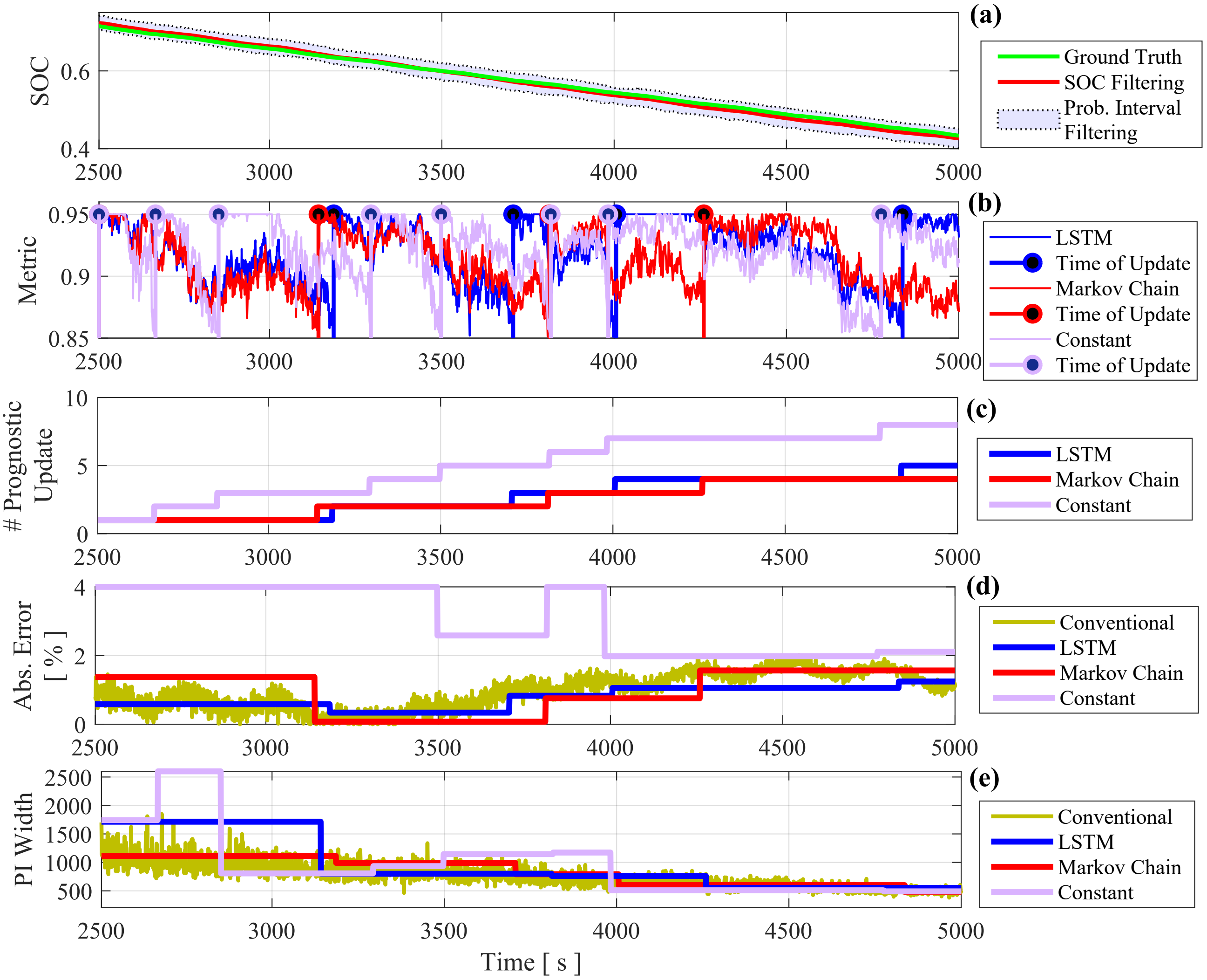

Fig. 1. Summary of results in terms of real-time prognostic performance metrics

Chain models exhibit similar behavior, which is close to the Conventional approach. Consequently, both models keep acceptable prognostic efficacy in terms of PI Width, even so, they demand significantly less computational resources. Furthermore, we note that PI Width is decreasing as time progresses. This phenomenon is expected, considering that the prognostic uncertainty increases as a function of the prognostic horizon. Thus, the shorter the prognostic horizon, the smaller the prognostic uncertainty.

At this point, it is also meaningful to discuss the training time of the models implemented to characterize future exogenous inputs. Measuring directly the computing time in Matlab, the training time for LSTM and Markov Chain were $1s$ and $0.1s$, respectively. It is not significant to consider the training time of the Constant model since the calculations are very basic. Considering these times, the first two methods are certainly more time-consuming at the training stage than the former. However, as previously noted, they produce more exact predictions as well as also less prognostic updates. Therefore, the extra time invested in training paid off at the prognostic stage. Nevertheless, the previous conclusion cannot be generalized because defining which model is the most adequate to characterize future operating conditions for prognostic purposes strongly depends on the signal shape. For instance, if the input is mostly constant, a model for characterizing future operating conditions that utilizes an average of the past n samples could be as effective as a more complex one that employs advanced tools, e.g., LSTM or Markov Chain. Hence, in such a case, a simple model might be the most recommendable alternative.

5. CONCLUSIONS

First of all, it is noteworthy that the model utilized to characterize future operating conditions plays a major role not only in the prognostics efficacy but also in the number of prognostic updates. On the one hand, as observed in the case of Constant model, an over-simplistic model could be easier to train, however, offers a poorer characterization of future operating conditions, which triggers two negative effects: (1) a reduction in the prognostics efficacy and (2) an increment in the number of prognostics updates required. On the contrary, as reviewed in the cases of LSTM and Markov Chain models, a complex model could be more time-demanding at the training stage but offers outstanding results for prognostic purposes. Nevertheless, if the exogenous input shape is simple (e.g., constant), both complex and simplistic models might have the same performance. Thus, prognostic designers must consider the signal shape in the definition of a model to characterize it.

Last but not least, this paper also highlights the performance of time-varying prognostics update procedure that is a feasible method to effectively reduce computational costs associated with the implementation of PFP algorithms. In fact, in the case of study, we should have needed 2500 prognostic executions under the conventional prognostic update approach. However, using time-varying updates, we compute less than 10 updates, while being as effective as the conventional approach.

REFERENCES

Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188.

Celaya, J.R., Saxena, A., and Goebel, K. (2012). Uncertainty representation and interpretation in model-based prognostics algorithms based on Kalman filter estimation. In *Annual Conference of the Prognostics and Health Management Society 2012*, 1–10.

Daigle, M. and Sankararaman, S. (2013). Advanced Methods for Determining Prediction Uncertainty in Model-Based Prognostics with Application to Planetary Rovers. In *Annual Conference of the Prognostics and Health Management Society 2013*, 13.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

Jouin, M., Gouriveau, R., Hissel, D., Péra, M.C., and Zerhouni, N. (2016). Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72, 2–31.

Kordestani, M., Saif, M., Orchard, M.E., Razavi-Far, R., and Khorasani, K. (2019). Failure prognosis and applications—a survey of recent literature. *IEEE transactions on reliability.*

Orchard, M. and Vachtsevanos, G. (2009). A particle-filtering approach for on-line fault diagnosis and failure prognosis. *Transactions of the Institute of Measurement and Control 2009*, 31, 221–246.

Pola, D.A., Navarrete, H.F., Orchard, M.E., Rabié, R.S., Cerda, M.A., Olivares, B.E., Silva, J.F., Espinoza, P.A., and Pérez, A. (2015). Particle-filtering-based discharge time prognosis for lithium-ion batteries with a statistical characterization of use profiles. *IEEE Transactions on Reliability*, 64(2), 710–720.

Rozas, H., Jaramillo, F., Perez, A., Jimenez, D., Orchard, M.E., and Medjaher, K. (2020). A method for the reduction of the computational cost associated with the implementation of particle-filter-based failure prognostic algorithms. *Mechanical Systems and Signal Processing*, 135, 106421.

Sankararaman, S. (2015). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing*, 52-53, 228–247.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing.* Bayesian Filtering and Smoothing. Cambridge University Press.

Sierra, G., Orchard, M., Goebel, K., and Kulkarni, C. (2019). Battery health management for small-size rotary-wing electric unmanned aerial vehicles: An efficient approach for constrained computing platforms. *Reliability Engineering & System Safety*, 182, 166–178.

Tamssaouet, F., Nguyen, K., Medjaher, K., and Orchard, M. (2019). Uncertainty quantification in system-level prognostics: Application to Tennessee Eastman Process. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1243–1248. IEEE.

Yan, W., Zhang, B., Wang, X., Dou, W., and Wang, J. (2015). Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries. *IEEE Transactions on Industrial Electronics*, 63(3), 1804–1812.