A new algorithm for prognostics using Subset Simulation

Manuel Chiachío, Juan Chiachío, Shankar Sankararaman,
Kai Goebel, John Andrews

PII: S0951-8320(16)30733-5
DOI: 10.1016/j.ress.2017.05.042
Reference: RESS 5862

To appear in: *Reliability Engineering and System Safety*

Received date: 7 November 2016
Revised date: 12 May 2017
Accepted date: 27 May 2017

Please cite this article as: Manuel Chiachío, Juan Chiachío, Shankar Sankararaman, Kai Goebel, John Andrews, A new algorithm for prognostics using Subset Simulation, *Reliability Engineering and System Safety* (2017), doi: 10.1016/j.ress.2017.05.042

Highlights

- A new algorithm based on Subset Simulation is provided for general prognostics;
- The Subset Simulation method is used to obtain efficiency for rare events;
- A simulated example and a challenging case study are used to demonstrate its efficacy;
- Discussion is provided through comparison with a standard prognostics algorithm;

A new algorithm for prognostics using Subset Simulation

Manuel Chiachío[a,*], Juan Chiachío[a], Shankar Sankararaman[b], Kai Goebel[c], John Andrews[a]

[a] *Resilience Engineering Research Group, University of Nottingham, Nottingham, NG7 2RD, UK*
[b] *SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035-1000*
[c] *NASA Ames Research Center, Intelligent Systems Division. Moffett Field, CA 94035-1000*

Abstract

This work presents an efficient computational framework for prognostics by combining the particle filter-based prognostics principles with the technique of Subset Simulation, first developed in S.K. Au and J.L. Beck [*Probabilistic Engrg. Mech., 16 (2001), pp. 263-277*], which has been named PFP-SubSim. The idea behind PFP-SubSim algorithm is to split the multi-step-ahead predicted trajectories into multiple branches of selected samples at various stages of the process, which correspond to increasingly closer approximations of the critical threshold. Following theoretical development, discussion and an illustrative example to demonstrate its efficacy, we report on experience using the algorithm for making predictions for the *end-of-life* and *remaining useful life* in the challenging application of fatigue damage propagation of carbon-fibre composite coupons using structural health monitoring data. Results show that PFP-SubSim algorithm outperforms the traditional particle filter-based prognostics approach in terms of computational efficiency, while achieving the same, or better, measure of accuracy in the prognostics estimates. It is also shown that PFP-SubSim algorithm gets its highest efficiency when dealing with rare-event simulation.

Keywords: Prognostics, rare events, Stochastic modeling, Subset Simulation

1. Introduction

Prognostics is a key technology which allows us to manage assets based on their state of health, as opposed to scheduling periodic inspection and maintenance activities based on statistics of mean-time-to-failure or similar information [1, 2]. In practice, prognostics uses information from health monitoring systems to determine the state of health of components so as to make *end-of-life* (EOL) and *remaining useful life* (RUL) predictions based on estimations of the time when specific critical thresholds will be exceeded [3, 4]. The potential of prognostics in positively contributing to safety and asset availability relies in its capacity to anticipate an anomalous or faulty condition. In particular, one of the challenging problems in prognostics is to accurately predict the EOL/RUL of systems whose faulty behavior is unlikely [5, 6]. These problems

*Corresponding author. e-mail: manuel.chiachio@nottingham.ac.uk
Tel:(+44)01159513944

can be encountered in practice when predicting the collapse of structures under fatigue degradation, catastrophic failures in nuclear plants, run-away conditions in batteries, etc. Due to the lack of data for such improbable events, model-based instead of data-driven prognostics frameworks have attracted significant attention in the Prognostic and Health Management (PHM) community for their ability to yield accurate predictions using a limited amount of data [7]. Model-based prognostics uses the underlying first principles on which the evolution of the fault indicator is based, thereby reducing the *lack of knowledge* uncertainty often present in prognostics [8, 9]. Several examples are found in the literature dealing with model-based prognostics frameworks for a widespread range of applications, like fatigue damage evolution in engineering materials [10, 11], failure of electronic components [12], aging of batteries [13, 14], to name but a few. Besides model uncertainty, other important source of uncertainty present in a typical prognostics problem is the uncertainty coming from the use of a specific prognostics algorithm [15]. Sampling-based methods (e.g. particle filters [PF]) [16, 17] are examples of algorithms used to efficiently approximate the probability density function (PDF) of the predicted system states through a limited set of discrete *particle paths*, representing sample trajectories of the system evolution in the state space [18]. Since multi-step ahead state estimation is required in prognostics [19], then the statistical uncertainty that arises from the approximation by particles is propagated in time leading to an increase of the final uncertainty for the EOL/RUL estimation [20]. This drawback can be exacerbated when reaching the failure threshold is a rare event under the model representing the system evolution, since the referred sample trajectories result in long paths of predicted states. Higher-density sampling-based methods may be employed achieving higher resolution for the EOL/RUL predictions, however it is at the expense of a higher computational effort. On the other hand, choosing a conservative failure threshold might constitute a pragmatic alternative although it results in discarding potential useful life. To alleviate this critical issue, some particular solutions have appeared in the literature to gain prediction accuracy while keeping computational cost at acceptable levels [21–23]. However, specialized computational frameworks to achieve the required EOL/RUL prediction accuracy still remains very limited for prognostics involving rare-event simulation.

In this work, a general prognostics algorithm is proposed based on the Subset Simulation method [24]. Subset Simulation is an efficient simulation framework which transforms the simulation of a rare-event into the simulation of a sequence of events of higher probabilities. This general aspect makes Subset Simulation applicable to a broad range of areas of science and engineering where the simulation of an unlikely event is required [25, 26]. The pivotal idea behind Subset Simulation in application to prognostics is to split the multi-step-ahead predicted states of the system into multiple branches of selected samples ("seeds") at selected stages of the process. These seeds provide the starting points for reproducing offsprings of predicted states, thus leading to a nested sequence of subsets which are adaptively obtained until the failure region is reached. For the higher subset, all the samples are closely distributed in the vicinity of the final threshold achieving higher resolution for the PDF of EOL/RUL. The resulting computational framework is named

PFP-SubSim and a first version was presented in a conference paper in [27]. The focus here is on enhancements to the methodological aspects behind PFP-SubSim as a general prognostics algorithm. In addition, insightful examples and extended results are provided.

The paper is organized as follows. Section 2 overviews the mathematical basis and computational aspects of prognostics. In Section 3, the Subset Simulation method is described and specialized for application to prognostics, after which the PFP-SubSim algorithm is presented. The efficiency of PFP-SubSim is illustrated in Section 4 using both a numerical example and also a case study. A discussion about the performance of PFP-SubSim algorithm in relation to the standard PF-based prognostics algorithm is also provided in Section 4. Section 5 provides concluding remarks.

2. Foundations of prognostics

Let $\{x_n\}_{n\geqslant 0}$ be a Markov process described by the state vector x_n taking values in a space denoted by $\mathcal{X} \subset \mathbb{R}^{n_x}$, and $\theta \in \mathbf{\Theta} \subset \mathbb{R}^{n_\theta}$ a set of uncertain model parameters. Let us define an augmented state $z_n \equiv (x_n, \theta) \in \mathcal{Z} = \mathcal{X} \times \mathbf{\Theta} \subset \mathbb{R}^{n_z = n_x + n_\theta}$ representing the overall stochastic process including model parameters θ. Let us now assume that the transition rule for the Markov process can be described as follows:

$$\frac{dz_n}{dn} = \Upsilon(z_n, u, v) \tag{1}$$

where $\Upsilon : \mathbb{R}^{n_u} \times \mathbb{R}^{n_z} \to \mathbb{R}^{n_z}$ is a possibly nonlinear function of the system state z_n along with a set $u \in \mathbb{R}^{n_u}$ of input parameters to the system (loadings, environmental conditions, operating conditions, etc.). The term $v \in \mathbb{R}^{n_z}$ refers to the model error which represents the difference between the actual system state z_n and the state predicted by the hypothesized model $\Upsilon(z_n, u, v)$. Except for very simple cases, the analytical continuous-time formulation of the stochastic process described above, is rarely practical to describe real-world problems. Instead, the state evolution is commonly investigated under a discrete-time approach by approximating the derivative in Equation 1 as a constant within any time interval, so that $z_n = z(n \cdot \Delta t + \tau) \, \forall \tau \in [0, \Delta t)$, where Δt is sufficiently small. Hence, Equation 1 can be discretized to a difference equation:

$$z_n = \Upsilon_n(z_{n-1}, u_n, v_n) \tag{2}$$

where the states $z_n = z(n \cdot \Delta t)$ are assumed to follow a hidden Markov process with transition probability density given by $p(z_n|z_{n-1}), n \in \mathbb{N}$. Noisy observations, denoted here by $y_n \in \mathbb{R}^{n_y}$, are added to the system and assumed to be conditionally independent given the states z_n. They are expressed as a function of the latent damage states z_n through a measurement function $\psi : \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ as follows:

$$y_n = \psi_n(z_n, u_n, w_n) \tag{3}$$

where $w_n = w(n \cdot \Delta t)$ denotes the measurement error. It is also assumed that the error terms v_n and w_n from Equations 2 and 3 are random variables instead of deterministic fixed-valued variables, and that they are distributed following specified probability models. Based on these probability models, the PDFs for the state transition equation and observation equation are prescribed (see [11, 28] for further insight).

In prognostics, the interest is mainly to efficiently obtain predictions about future states of the system whereby EOL/RUL estimations are subsequently derived, provided that a failure region has been defined [18]. The predictions about future states are carried-out using no additional evidence but a sequence of measurements up to time n, denoted by $y_{0:n} \triangleq (y_0, y_1, \ldots, y_{n-1}, y_n)$. Two main steps are required for prognostics:

(i) *State estimate*: An estimate of the sequence of z-states $z_{0:n} \triangleq (z_0, z_1, \ldots, z_{n-1}, z_n)$ is first required, which is denoted by the PDF $p(z_{0:n}|y_{0:n})$. Here, the conditioning on $y_{0:n}$ is to indicate that predictions are based upon information using most available measurements up to time n. This *updated* PDF is given by Bayes' theorem as follows [1, 2]:

$$p(z_{0:n}|y_{0:n}) = \frac{p(y_n|z_n)p(z_{0:n}|y_{0:n-1})}{\int_{\mathcal{Z}} p(y_n|z_n)p(z_{0:n}|y_{0:n-1})dz_{0:n}} \propto p(y_n|z_n)p(z_n|z_{n-1}) \underbrace{p(z_{0:n-1}|y_{0:n-1})}_{\text{last update}} \tag{4}$$

where

$$p(z_n|z_{n-1}) = p(x_n|x_{n-1}, \theta_n)p(\theta_n|\theta_{n-1}) \tag{5}$$

In Equation 4, the identities $p(y_n|z_{0:n}, y_{0:n-1}) = p(y_n|z_n)$ and $p(z_n|z_{0:n-1}, y_{0:n-1}) = p(z_n|z_{n-1})$ are assumed based on the definition of the measurement equation (recall Equation 3) and the Markovian property of the state transition equation, respectively. It is also assumed that the initial state z_0 is known in advance, hence $p(z_0|y_0) \equiv p(z_0)$ (note that y_0 is not a measurement), being $p(z_0)$ the prior PDF of the system state. Moreover, as observed from Equation 5, model parameters θ_n are assumed to evolve by some unknown random process that is independent of the system state x_n. It is, in fact, a key problem that typically arises when sequentially updating the state vector $z_{0:n} = (x_{0:n}, \theta)$ as an augmented state due to the non-dynamics nature of θ. A common solution is to add a small random perturbation to θ (e.g., zero-mean Gaussian) under the last posterior PDF at time $n-1$ before evolving to the next predicted state at time n [29], i.e.:

$$\theta_n \sim p(\theta|\theta_{n-1}) = \mathcal{N}(\theta_{n-1}, W_n) \tag{6}$$

where $W_n \in \mathbb{R}^{n_\theta \times n_\theta}$ is a specified covariance matrix. Observe that by this method, the model parameters are virtually time-evolving although they are essentially not dependent on time. This time-varying imposes a loss of information in θ over time as additional uncertainties are artificially added to the parameters, which ultimately influence the precision of the filtering. There exist several methods in the literature to overcome this drawback, with the most popular being those that impose some kind of

shrinkage over W_n as long as new data are collected. The most important development in this direction is provided by Liu and West [30], whilst new methods in the context of prognostics have been recently proposed by [23, 31].

(ii) *Failure prediction*: Having estimated the latest available state updated at n, the next steps for prognostics are: a) to predict the distribution of future states of the system ℓ-steps forward in time in absence of new observations, i.e., $p(z_{n+\ell}|y_{0:n})$, where $\ell > 1$, b) to scrutinize whether the predicted state $z_{n+\ell}$ has reached the failure region. The ℓ-step ahead prediction is accomplished by Total Probability theorem as [32]:

$$\begin{aligned} p(z_{n+\ell}|y_{0:n}) &= \int_{\mathcal{Z}} p(z_{n+\ell}|z_{n:n+\ell-1}, y_{0:n}) p(z_{n:n+\ell-1}|y_{0:n}) dz_{n:n+\ell-1} \\ &= \int_{\mathcal{Z}} \left[\prod_{t=n+1}^{n+\ell} p(z_t|z_{t-1}) \right] p(z_n|y_{0:n}) dz_{n:n+\ell-1} \end{aligned} \tag{7}$$

where $p(z_n|y_{0:n})$ is the PDF which provides us with the up-to-date information about the system at time n. In the last equation, the identity $p(z_{n+\ell}|z_{n:n+\ell-1}) = p(z_{n+\ell}|z_{n+\ell-1})$ holds, since $\{z_n\}_{n\in\mathbb{N}}$ defines a Markov model of order one and also by the assumption that the observations are conditionally independent given the states.

Next, to perform failure prediction, a definition of a failure region is first required. To this end, it is denoted by $\mathcal{U} \subset \mathcal{Z}$ the non-empty subset of "authorized" states of our system, and the complementary subset $\bar{\mathcal{U}} = \mathcal{Z} \setminus \mathcal{U}$, the subset of states where the system behavior becomes unacceptable, or simply, where system failure occurs. The interest is on making predictions of the earliest time when the system failure occurs, i.e., the time $n + \ell$ so that the state $z_{n+\ell}$ predicted according to Equation 1 lies inside $\bar{\mathcal{U}}$, whereby the EOL/RUL can be obtained as Figure 1 illustrates. See [1, 2, 10] for further details.

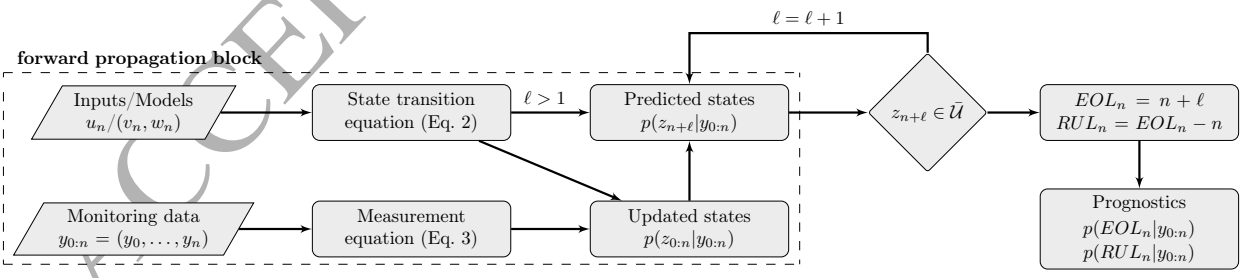

Figure 1: Conceptual scheme of a filtering-based prognostics framework with indication of RUL/EOL calculation.

2.1. Sequential Monte Carlo for state estimation

The recursive scheme for prognostics explained above is only a theoretical solution since, in general, the integrals involved in the Total Probability theorem and Bayes' theorem cannot be calculated analytically, except for some of especial linear cases using Gaussian uncertainties [17]. An alternative for the general case of non-linear and/or non-Gaussian state-space models is to use *particle methods* [33], a set of sequential Monte Carlo methods which provide samples (*particles*) approximately distributed according to a specific target PDF, with a feasible computational burden. Particle filters (PF) [29] are one of the most common techniques among particle methods for filtering [16] and prognostics [34]. With PF, the approximation of the state distribution $p(z_{0:n}|y_{0:n})$ is described through a set of N discrete particle paths, namely $\{z_{0:n}^{(i)} = (x_{0:n}^{(i)}, \theta_{0:n}^{(i)})\}_{i=1}^{N}$, which are readily sampled from a convenient importance distribution $q(z_{0:n}|y_{0:n})$ as follows:

$$p(z_{0:n}|y_{0:n}) \approx \sum_{i=1}^{N} \hat{\omega}_n^{(i)} \delta(z_{0:n} - z_{0:n}^{(i)}) \tag{8}$$

where δ is the Dirac delta and $\hat{\omega}_n^{(i)}$ is the unnormalized importance weight for the i^{th} particle, which can be obtained as:

$$\hat{\omega}_n^{(i)} = \frac{p(z_{0:n}^{(i)}|y_{0:n})}{q(z_{0:n}^{(i)}|y_{0:n})} \tag{9}$$

Methods for choosing the importance density are well known in the literature, hence they are not repeated here [30, 35]. However in most of the applications, the importance density is conveniently chosen as $q(z_{0:n}|y_{0:n}) = q(z_{0:n}|y_{0:n-1})$, so that it admits a sample procedure since it can be factorized in a form similar to that of the target posterior PDF, i.e., $q(z_{0:n}|y_{0:n}) = q(z_{0:n-1}|y_{0:n-1})q(z_n|z_{n-1})$. By substituting Equation 4 into Equation 9 and also by using the last cited condition for the importance PDF, then the unnormalized importance weight for the i^{th} particle at time n can then be rewritten as:

$$\hat{\omega}_n^{(i)} \propto \underbrace{\frac{p(x_{0:n-1}^{(i)}, \theta_{0:n-1}^{(i)}|y_{0:n-1})}{q(x_{0:n-1}^{(i)}, \theta_{0:n-1}^{(i)}|y_{0:n-1})}}_{\hat{\omega}_{n-1}^{(i)}} \frac{p(x_n^{(i)}|x_{n-1}^{(i)}, \theta_{n-1}^{(i)}) p(y_n|x_n^{(i)}, \theta_n^{(i)})}{q(x_n^{(i)}|x_{n-1}^{(i)}, \theta_n^{(i)})} \tag{10}$$

where the joint state z_n has been decomposed into its two components (x_n, θ) for better comprehension of the weight calculation method. Guidelines on how to best select $q(x_n|x_{n-1}, \theta_{n-1})$ can be found in [16], and an often followed method is to use the conditional distribution $p(x_n|x_{n-1}, \theta_{n-1})$ from the transition equation since it is straightforward to evaluate [17]. By means of this, the expression for the i^{th} unnormalized particle weight yields:

$$\hat{\omega}_n^{(i)} = \hat{\omega}_{n-1}^{(i)} p(y_n|z_n^{(i)}) = \hat{\omega}_{n-1}^{(i)} p(y_n|x_n^{(i)}, \theta_n^{(i)}) \tag{11}$$

Observe from last equation that the weight values are known only up to a scaling factor, which can be readily bypassed by normalization, i.e. $\omega_n^{(i)} = \hat{\omega}_n^{(i)} / \sum_{i=1}^{N} \hat{\omega}_n^{(i)}$, where $\omega_n^{(i)}$ denotes the normalized value of the i^{th} particle at time n. A pseudocode implementation for the PF is given as Algorithm 1. Note that a

systematic resampling step has been included in Algorithm 1 to limit the well-known degeneracy problem of the PF during resampling [36]. In PF, particles are either dropped or reproduced during resampling which may result in a loss of diversity of particle paths [16]. A control step on this degeneracy by using the effective sample size (ESS) [37] may be incorporated before the resampling.

2.2. Particle filtering-based prognostics

Using the PF approach presented above, the EOL predicted at time n is obtained for the i^{th} particle trajectory as the time index of the first-passage point into $\bar{\mathcal{U}}$, or in other words, the earliest time index $n+\ell, \ell \geqslant 1$ so that the event $z_{n+\ell}^{(i)} \in \bar{\mathcal{U}}$ occurs. It can be computed as:

$$\mathrm{EOL}_n^{(i)} = \inf\left\{n+\ell \in \mathbb{N} : \ell \geqslant 1 \wedge \mathbb{I}_{(\bar{\mathcal{U}})}(z_{n+\ell}^{(i)}) = 1\right\} \tag{12}$$

where $\mathbb{I}_{(\bar{\mathcal{U}})} : \mathcal{Z} \to \{0,1\}$ is an indicator function that maps a given point in $\mathcal{Z}$ to the Boolean domain $\{0,1\}$ as follows:

$$\mathbb{I}_{(\bar{\mathcal{U}})}(z) = \begin{cases} 1, & \text{if } z \in \bar{\mathcal{U}} \\ 0, & \text{if } z \in \mathcal{U} \end{cases} \tag{13}$$

Algorithm 1 PF with on-line parameter updating

Inputs: N, {number of particles per time step}, $N^{'}$, {threshold of effective sample size (ESS)}, $\left\{z_0^{(i)} = \left(x_0^{(i)}, \theta_0^{(i)}\right), \omega_0^{(i)}\right\}_{i=1}^{N}$ {initial particles from prior PDF $p(z_0)$}

Outputs: $\left\{z_n^{(i)} = \left(x_n^{(i)}, \theta_n^{(i)}\right), \omega_n^{(i)}\right\}_{i=1}^{N}$, {updated particles at time n}

Begin ($n \geqslant 1$): {time n evolves as new data arrive}

for $i = 1, \ldots, N$ **do**
  Sample $\theta_n^{(i)} \sim p(\theta_n | \theta_{n-1}^{(i)})$ {e.g., use the Liu & West method [30]}
  Sample $x_n^{(i)} \sim p(x_n | x_{n-1}^{(i)}, \theta_n^{(i)})$
  $z_n^{(i)} \leftarrow (x_n^{(i)}, \theta_n^{(i)})$ and $z_{0:n}^{(i)} \leftarrow (x_{0:n}^{(i)}, \theta_{0:n}^{(i)})$
  $\hat{\omega}_n^{(i)} \leftarrow \omega_{n-1}^{(i)} p(y_n | z_n^{(i)})$
end for
for $i = 1, \ldots, N$ **do**
  $\omega_n^{(i)} \leftarrow \frac{\hat{\omega}_n^{(i)}}{\sum_{i=1}^{N} \hat{\omega}_n^{(i)}}$ {normalize weights}
end for
if EES $< N^{'}$ **then**
  $\left\{z_n^{(i)}, \omega_n^{(i)}\right\}_{i=1}^{N} \leftarrow \texttt{resample}\left\{z_n^{(i)}, \omega_n^{(i)}\right\}_{i=1}^{N}$
end if

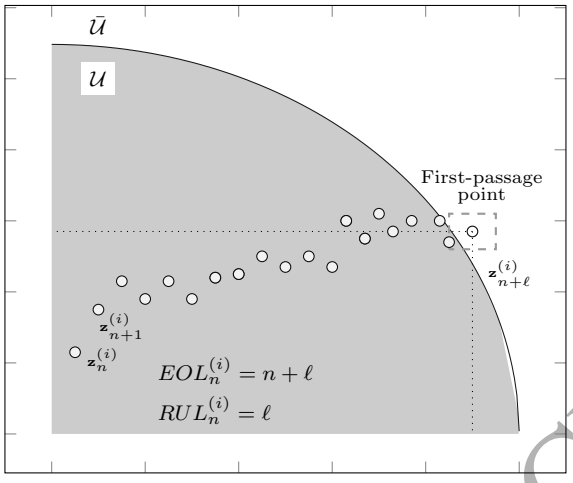

Figure 2: Conceptual illustration of the i^{th} particle trajectory to EOL. Solid disks represent samples in $\mathcal{Z}$.

The RUL predicted from time n for the i^{th} particle trajectory can be straightforwardly obtained from $\text{EOL}_n^{(i)}$ as $\text{RUL}_n^{(i)} = \text{EOL}_n^{(i)} - n$. An approximation to the PDF of EOL at time n can be obtained as:

$$p(\text{EOL}_n|y_{0:n}) \approx \sum_{i=1}^{N} \omega_n^{(i)} \delta(EOL_n - EOL_n^{(i)}) \tag{14}$$

Analogously, the PDF of RUL_n is approximated as:

$$p(\text{RUL}_n|y_{0:n}) \approx \sum_{i=1}^{N} \omega_n^{(i)} \delta(RUL_n - RUL_n^{(i)}) \tag{15}$$

Figure 2 provides a schematic illustration to exemplify the trajectory of the i^{th} particle of a general z-state along with the indication of $\text{EOL}_n^{(i)}$ and $\text{RUL}_n^{(i)}$. An algorithmic description of the particle filtering-based prognostics procedure is provided as Algorithm 2.

3. Prognostics using Subset Simulation

3.1. Basis of Subset Simulation

Subset Simulation method is an efficient simulation framework originally proposed for computing small failure probabilities for general reliability problems [24]. It is motivated by the observation that the simulation of a rare event can be transformed into the simulations of successive intermediate events with larger probabilities. The small probability of the rare event can then be expressed as a product of larger conditional probabilities that can be obtained with much less computational effort. In Subset Simulation, the conditional probabilities are efficiently estimated by means of conditional samples that correspond to specified levels of

Algorithm 2 Standard PF-prognostic algorithm

Inputs: $\left\{z_n^{(i)} = \left(x_n^{(i)}, \theta_n^{(i)}\right), \omega_n^{(i)}\right\}_{i=1}^{N}$ {updated particles at time n. Use Algorithm 1}, $\bar{\mathcal{U}} \subset \mathcal{Z}$ {failure domain},

Outputs: $EOL_n,\ RUL_n$

Begin ($t \geqslant n$):

for $i = 1, \ldots, N$ **do**
  $t \leftarrow n$
  $z_t^{(i)} \leftarrow z_n^{(i)}$
  Evaluate $EOL_n^{(i)}(z_t^{(i)})$, {use Eq. 12 & 13}
  while $\mathbb{I}_{(\bar{\mathcal{U}})}(z_t^{(i)}) = 0$ **do**
    Sample $\theta_{t+1}^{(i)} \sim p(\theta_{t+1}|\theta_t^{(i)})$
    Sample $x_{t+1}^{(i)} \sim p(x_{t+1}|x_t^{(i)}, \theta_t^{(i)})$
    $t \leftarrow t + 1$
    $z_t = \left(x_t^{(i)}, \theta_t^{(i)}\right) \leftarrow z_{t+1} = \left(x_{t+1}^{(i)}, \theta_{t+1}^{(i)}\right)$
  end while
  $EOL_n^{(i)} \leftarrow t,\ RUL_n^{(i)} = EOL_n^{(i)} - n$
end for

a performance function $g : \mathcal{Z} \to \mathbb{R}$ in a progressive manner. Let us assume that, with no loss of generality, $\bar{\mathcal{U}}$ can be expressed through evaluation of exceedance of the performance function g above some specified threshold level b, as follows:

$$\bar{\mathcal{U}} \triangleq \{z \in \mathcal{Z} : g(z) > b\} \tag{16}$$

Let us also assume that the failure region $\bar{\mathcal{U}}$ is defined as the intersection of m nested regions in $\mathcal{Z}$, i.e., $\bar{\mathcal{U}}_1 \supset \bar{\mathcal{U}}_2 \ldots \supset \bar{\mathcal{U}}_{m-1} \supset \bar{\mathcal{U}}_m = \bar{\mathcal{U}}$, so that $\bar{\mathcal{U}} = \bigcap_{j=1}^{m} \bar{\mathcal{U}}_j$. Each subset $\bar{\mathcal{U}}_j$ is typically termed as *intermediate failure domain* and can be defined as[1] $\bar{\mathcal{U}}_j \triangleq \{z \in \mathcal{Z} : g(z) > b_j\}$, with $b_{j+1} > b_j$. Note that when $\bar{\mathcal{U}}_j$ holds, then $\{\bar{\mathcal{U}}_{j-1}, \ldots, \bar{\mathcal{U}}_1\}$ also hold, and hence $P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}, \ldots, \bar{\mathcal{U}}_1) = P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1})$, so it follows that[2]:

$$P(\bar{\mathcal{U}}) = P\Big(\bigcap_{j=1}^{m} \bar{\mathcal{U}}_j\Big) = P(\bar{\mathcal{U}}_1) \prod_{j=2}^{m} P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}) \tag{17}$$

where $P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}) \equiv P(z \in \bar{\mathcal{U}}_j | z \in \bar{\mathcal{U}}_{j-1})$, the conditional probability for the $(j-1)^{th}$ intermediate failure domain, which is denoted by P_j hereinafter for simplicity. Equation 17 indicates that the probability $P(\bar{\mathcal{U}})$ may be relatively small, however it can be approximated by Subset Simulation as the product of larger conditional probabilities, thus avoiding simulation of rare events.

[1] It is assumed that z_n describes a monotonically increasing process. In the contrary case, $\bar{\mathcal{U}}_j \equiv \{z \in \mathcal{Z} : g(z) < b_j\}$, where $b_{j+1} < b_j$.

[2] In what follows, we use $P(\cdot)$ to denote probability whereas a PDF is expressed as $p(\cdot)$.

3.2. Using Subset Simulation for prognostics

In this section, Subset Simulation method is exploited as an efficient sampler for prognostics involving rare-event simulation. As previously mentioned in Section 2, a prognostics algorithm aims at obtaining simulations of the model within the failure region whereby EOL/RUL estimations are subsequently derived. Let $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}) \propto p(z_{n:n+\ell}|y_{0:n})\mathbb{I}_{\bar{\mathcal{U}}}(z_{n:n+\ell})$ denotes the PDF of ℓ-step ahead predicted states distributed within the region $\bar{\mathcal{U}}$, so that $p(z_{n:n+\ell}|y_{0:n})$ represents the probability model for predicted z-states for the interval $(n:n+\ell]\,, \ell \geqslant 1$, which can be obtained by Total Probability theorem using the state transition equation $p(z_t|z_{t-1})$ (see [2, 11] for further insights about predicting future states from the transition equation). The probability of expected performance of the predicted z-states within $\bar{\mathcal{U}}$ can be obtained as a probability integral, i.e., $P(\bar{\mathcal{U}}) = \int_{\bar{\mathcal{U}}} p(z_{n:n+\ell}|y_{0:n})dz_{n:n+\ell}$, which can be approximated using Subset Simulation as a product of conditional probabilities as stated by Equation 17. The term P_1 from Equation 17 can be readily estimated by the standard Monte Carlo method (MC) as follows:

$$P(\bar{\mathcal{U}}_1) \approx \bar{P}_1 = \frac{1}{M}\sum_{k=1}^{M} \mathbb{I}_{\bar{\mathcal{U}}_1}\left(z_{n:n+\ell}^{1,(k)}\right) \tag{18}$$

where $\left\{z_{n:n+\ell}^{1,(k)}\right\}_{k=1}^{M}$ are M samples simulated according to the PDF $p(z_{n:n+\ell}|y_{0:n})$. The superscript "1" here indicates that such samples lie within the region $\bar{\mathcal{U}}_1$. The remaining factors can be efficiently estimated when $j \geqslant 2$ by conditional sampling from $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_{j-1}) \propto p(z_{n:n+\ell}|y_{0:n})\mathbb{I}_{\bar{\mathcal{U}}_{j-1}}(z_{n:n+\ell})$, giving:

$$P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}) \approx \bar{P}_j = \frac{1}{M}\sum_{k=1}^{M} \mathbb{I}_{\bar{\mathcal{U}}_j}\left(z_{n:n+\ell}^{j-1,(k)}\right) \tag{19}$$

where $z_{n:n+\ell}^{j-1,(k)} \sim p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_{j-1})$, and $\mathbb{I}_{\bar{\mathcal{U}}_j}(z_{n:n+\ell}^{j-1,(k)})$ is an indicator function for the region $\bar{\mathcal{U}}_j, j = 1,\ldots,m$, that assigns a value of 1 when $g(z_{n:n+\ell}^{j-1,(k)}) > b_j$, and 0 otherwise. Note that, by virtue of the Markov property of the state-space model, $p(z_{n:n+\ell}|y_{0:n})$ can be approximated by conditional sampling using recursively the one-step transition equation $p(z_n|z_{n-1})$ [2], i.e.: first sample $z_{n+1}^{(k)}$ using the recurrence given by the one-step transition equation conditional on the state z_n, i.e., $z_{n+1}^{(k)} \sim p(\cdot|z_n)$; then sample the succeeding state conditional on the previous sample, i.e., $z_{n+2}^{(k)} \sim p(\cdot|z_{n+1}^{(k)})$; finally, repeat the same process until the time index $n+\ell$ has been reached. Observe that in Subset Simulation for prognostics, there is no need to invoke Markov chain Monte Carlo methods [38] for conditional sampling as was originally proposed in [24], since conditional samples are straightforwardly obtained by simulating the state transition equation, as has just been explained above.

Note also that it is possible to obtain samples that are generated at the $(j-1)^{th}$ level which lie in the subsequent level $\bar{\mathcal{U}}_j$. They are samples conditional on $\bar{\mathcal{U}}_j$ and provide "seeds" for simulating more samples according to $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_j)$ [39]. Moreover, in Subset Simulation the choice of intermediate levels can be addressed by adaptively specifying the threshold value $b_j, j = 1,\ldots,m$, as the $[MP_0]^{th}$ largest value

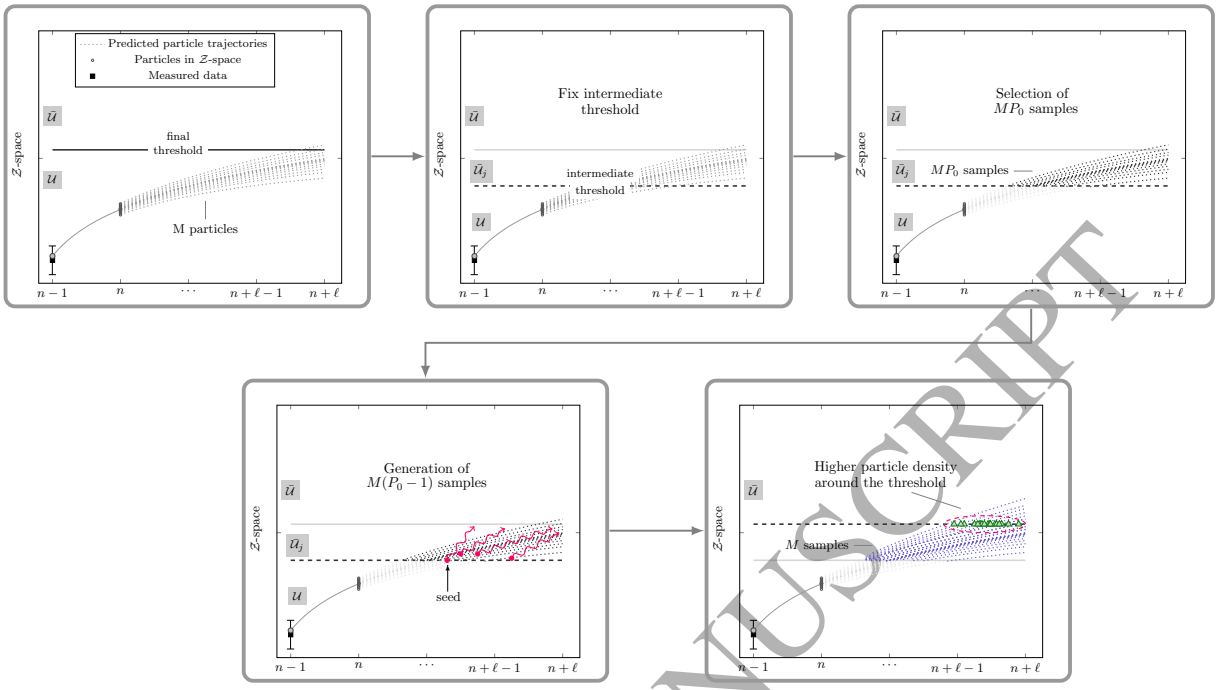

Figure 3: Schematic representation of conditional samples produced using PFP-SubSim algorithm.

among the values $g(z_{n:n+\ell}^{j-1,(k)})$, so that the sample estimate of $P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1})$ in Equation 19 is equal to a fixed value $P_0 \in (0,1)$. Therefore, $\bar{\mathcal{U}}_j$ is adaptively chosen based on the samples $\{z_{n:n+\ell}^{j-1,(k)}\}_{k=1}^{M}$ generated from $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_{j-1})$, in such a way that there are exactly MP_0 of these samples in $\bar{\mathcal{U}}_j$, which serve as seeds for generating more samples according to $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_j)$ [39, 40]. In fact, the remaining $(1/P_0 - 1)$ samples are generated from $p(z_{n:n+\ell}|y_{0:n},\bar{\mathcal{U}}_j)$ by simulating the state space model starting at each seed, giving a total of M samples in $\bar{\mathcal{U}}_j$. This procedure is repeated for higher conditional levels until the final region $\bar{\mathcal{U}}_m = \bar{\mathcal{U}}$ has been reached. The PDF of EOL/RUL can be straightforwardly obtained by applying Equations 12 to 15 using as samples those distributed within the final region $\bar{\mathcal{U}}_m$. A schematic representation of the main steps of the proposed Subset Simulation approach has been provided in Figure 3. These steps can be summarized as follows (the sequence is indicated by the arrows in Figure 3): (i) generation of ℓ-step ahead predictive samples from $p(z_n|y_{0:n})$; (ii) adaptive fixing of the intermediate threshold value b_j; (iii) definition of the intermediate region $\bar{\mathcal{U}}_j$ so that $P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}) = P_0$; (iv) generation of new samples distributed according to $p(\cdot|y_{0:n},\bar{\mathcal{U}}_j)$, $j = 1,\ldots,m$, (v) calculation of EOL/RUL using the M samples from the final subset.

3.3. The PFP-SubSim algorithm

A pseudocode implementation of PFP-SubSim algorithm is provided as Algorithm 3, which is intended to be sufficient for most cases of application. With no loss of generality, the pseudocode is implemented such

that a fixed amount of M samples are drawn per simulation level $\bar{\mathcal{U}}_j$, so that $N_T = mM$, the total amount of model evaluations required by the algorithm to reach the final threshold. It is important to remark that this choice is just to allow the computational cost to be controlled.

Algorithm 3 Pseudocode implementation for PFP-SubSim

Inputs: $P_0 \in (0,1)$ {gives percentile selection, chosen so $NP_0, {}^1/_{P_0} \in \mathbb{Z}^+$; $P_0 = 0.2$ is recommended}, M {number of samples per simulation level}, $b \in \mathbb{R}$ {threshold value}, $\left\{z_n^{(i)} = \left(x_n^{(i)}, \theta_n^{(i)}\right), \omega_n^{(i)}\right\}_{i=1}^{N}$ {e.g. use Algorithm 1}

Outputs: $EOL_n,\ RUL_n$

Begin:

Sample $\left(z_n^{(1)}, \ldots, z_n^{(N)}\right)$, according to weights $\left\{\omega_n^{(i)}\right\}_{i=1}^{N}$
Set $k = 0$, $j = 0$
for $i : 1, \ldots, N$ **do**
  $t \leftarrow n$
  $z_t^{(i)} \leftarrow z_n^{(i)}$
  repeat
    Sample $z_{t+1}^{(i)} \sim p(z_{t+1}|z_t^{(i)})$
    $t \leftarrow t + 1$
    $k \leftarrow k + 1$
    $z_t^{(i)} \leftarrow z_{t+1}^{(i)}$
    $z^{j,(k)} \leftarrow z_t^{(i)}$
    Evaluate $g_j^{(k)} = g\left(z^{j,(k)}\right)$
  until $t = n + \ell,\ \ell = {}^M/_N$
end for
Sort $\left\{z^{j,(k)}\right\}_{k=1}^{M}$ so that $g_j^{(1)} \leqslant g_j^{(2)} \leqslant \ldots \leqslant g_j^{(M)}$ $\quad \{g_j^{(1)} \geqslant g_j^{(2)} \geqslant \ldots \geqslant g_j^{(M)}$ for decreasing processes$\}$
$b_j \leftarrow \frac{1}{2}\left(g_j^{(MP_0)} + g_j^{(MP_0+1)}\right)$
while $b_j < b$ {Set $b_j > b$ for decreasing processes} **do**
  $j \leftarrow j + 1$
  for $k = 1, \ldots, MP_0$ **do**
    Select as a seed $\left(z^{j,(k),(1)}\right) = \left(z^{j-1,(k)}\right) \sim p\left(z|\bar{\mathcal{U}}_j\right)$
    Sample from Eq. (5) to generate $1/P_0$ states of a Markov chain lying in $\bar{\mathcal{U}}_j$: $\left(z^{j,(k),(1)}, \ldots, z^{j,(k),(1/P_0)}\right)$
  end for
  Renumber $\left\{z^{j,(k),(i)}\right\}_{k=1,i=1}^{MP_0,1/P_0}$ as $\left(z^{j,(1)}, \ldots, z^{j,(M)}\right)$
  Sort $\left\{z^{j,(k)}\right\}_{k=1}^{M}$ so that $g_j^{(1)} \leqslant g_j^{(2)} \leqslant \ldots \leqslant g_j^{(M)}$ $\quad \{g_j^{(1)} \geqslant g_j^{(2)} \geqslant \ldots \geqslant g_j^{(M)}$ for decreasing processes$\}$
  $b_j \leftarrow \frac{1}{2}\left(g_j^{(MP_0)} + g_j^{(MP_0+1)}\right)$
end while
$m \leftarrow j$
Evaluate $\left\{EOL_n^{(k)}\right\}_{k=1}^{M}, \left\{RUL_n^{(k)} = EOL_n^{(k)} - n\right\}_{k=1}^{M}$, {use Eq. 12 & 13}

In the proposed algorithm, the choice of a suitable P_0 value has a significant impact, since a small value ($P_0 \to 0$) makes the distance between consecutive intermediate levels $b_j - b_{j-1}$ to become too large, which leads to a rare-event simulation problem [39, 41]. If, on the contrary, the intermediate threshold values were chosen too close ($P_0 \to 1$), the algorithm would require a large total number of simulation levels m (and hence high computational effort) to reach the target region $\bar{\mathcal{U}}$. A rational choice for P_0 should strike a balance between both extreme cases. In the original presentation of Subset Simulation in [24], $P_0 = 0.1$ was recommended, and in [39], the range $0.1 \leqslant P_0 \leqslant 0.3$ was found to be near optimal after a rigorous sensitivity study of Subset Simulation. More recently, in [40] the value $P_0 = 0.2$ was found to be optimal for Subset Simulation in application to Approximate Bayesian Computation problems. Following the recommendation by [39, 40], the conditional probability P_0 is conveniently chosen here so that MP_0 and $1/P_0$ are positive integers, thus P_0 is set to 0.2.

In Algorithm 3, the time subscripts are dropped from step 12 for simplicity, since the time indexing is prescribed in each sample. Observe that the proposed methodology, and in particular the steps 20 and 21 in Algorithm 3, have anticipated that the multi-step ahead predicted trajectories simulated according to the model (Equation 2) can be split into seeds whereby subsequent states are generated, without artificially influencing the recurrence given by the stochastic process. This is justified by the Markovian assumption, so that the probability of obtaining $z_{n+\ell}$ depends only on its preceding state $z_{n+\ell-1}$ and not on the history of past states. The last implies that the simulation of a sequence of states $\ell-$step ahead is essentially an uncoupled procedure given the information from the previous step.

4. Illustrative examples

4.1. Toy example

Consider an exponential degradation process described by the following discrete state-transition equation:

$$x_n = e^{-2\zeta n} x_{n-1} + v_n \tag{20}$$

where $x_n \in \mathbb{R}$ are discrete system states for $n \in \mathbb{N}$, $\zeta \in \mathbb{R}$ is the decay parameter, a scaling constant that controls the degradation velocity, and $v_n \in \mathbb{R}$ is the model error term which is assumed to be modeled as a zero-mean Gaussian distribution, i.e. $v_n \sim \mathcal{N}(0, \sigma_v)$. Let us now assume that the degradation process can be measured over time and that, at a certain time n, the measured degradation can be expressed as a function of the latent damage state x_n, as follows:

$$y_n = x_n + w_n \tag{21}$$

where $w_n \in \mathbb{R}$ is the measurement error, which is also assumed to be modeled as a zero mean Gaussian PDF with standard deviation given by σ_w. In this example, the standard deviations of the error terms

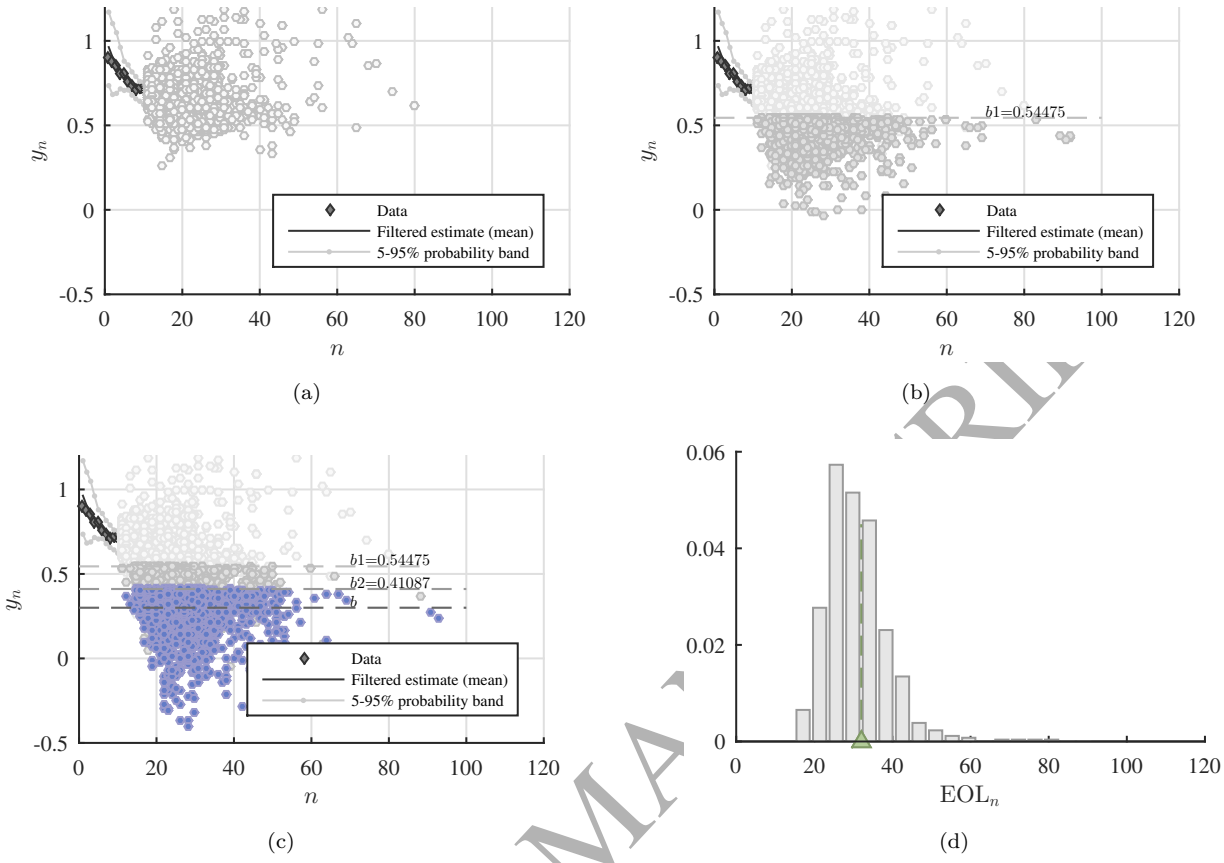

Figure 4: PFP-SubSim output for the exponential degradation model in Eq. 20 by using $\zeta = 0.015$. Predicted samples are represented in the state-space for $n+\ell, n = 10, \ell \geqslant 1$ using circles.

are selected as model parameters, i.e. $\theta = (\sigma_v, \sigma_w) \in \mathbb{R}^2$. The uniform PDF $\mathrm{U}(0.0001, 0.2)$ is used as component-wise prior information about the model parameters $\theta_1 = \sigma_v$ and $\theta_2 = \sigma_w$. Degradation states are initialized at $x_0 = 0.9$, expressed in arbitrary units. Synthetic data for y_n is used by generating them from Equations 20 and 21 considering $\theta_{\text{true}} = (0.01, 0.02)$. The PFP-SubSim results predicted at time $n = 10$ are presented in Figure 4 for three different simulation levels ($m = 3$) by using $P_0 = 0.2$. A total amount of $N = 1000$ particles trajectories are employed by Algorithm 1 (in this example, the resampling step is run every time new data are collected). A zero-mean Gaussian has been used for the artificial evolution of model parameters, i.e. $p(\theta_n|\theta_{n-1}) \sim \mathcal{N}(0, W_n)$, where $W_n = 5 \cdot 10^{-4} I_2$, being I_2 the identity matrix of order 2. Finally, the failure region is defined by fixing a threshold value for the states of 0.3, expressed in arbitrary units, i.e. $\bar{\mathcal{U}} = \{x_n \in \mathbb{R} : x_n < 0.3\}$.

In panel 4a, predicted samples are represented in the state-space for $n+\ell, n = 10, \ell \geqslant 1$ using circles. In panels 4b and 4c, samples are increasingly distributed in subsets according to $p(z_{n:n+\ell}|y_{0:n}, \bar{\mathcal{U}}_j)$, $j = 1, 2$. They are superimposed using increasing gray tones until the final region is reached (dark purple circles). Panel

ζ (decay parameter)	0.015	0.01	0.0075	0.005
Algorithm 2	42	76	95	150
PFP-SubSim	43	44	47	55
Efficiency factor	0.98	1.73	2.02	2.72

Table 1: Results of the comparative exercise between PFP-SubSim and Algorithm 2 for the exponential decay model of Eq. 20.

4d shows the histogram representation of the PDF $p(EOL_n|y_{0:n})$ predicted at time $n = 10$. The triangle represents the time when the threshold value $b = 0.3$ is reached according to the synthetic data. Observe that by using PFP-SubSim algorithm, the ℓ-step ahead predicted samples are distributed in subsets which are increasingly closer to the final threshold. The final subset (represented using dark purple circles in Figure 4c) encloses a considerable amount of samples distributed around the vicinity of the threshold, hence increasing the resolution of the PDF of EOL without the need of employing big amounts of model evaluations that unnecessarily increases the computational cost.

A comparative study has been carried-out to reveal the computational efficiency that can be gained using PFP-SubSim against Algorithm 2 for obtaining the PDF of EOL. For this exercise, the amount of model evaluations needed to produce one sample simulated according to the PDF $p(EOL_n|y_{0:n})$ is monitored at $n = 10$. In other words, the focus is on the ratio defined by the amount of samples that describe $p(EOL_n|y_{0:n})$ over the total amount of model evaluations employed by each of the algorithms used for comparison. The results, shown in the second and third rows of Table 1, are obtained by considering different values for the decay parameter ζ in Equation 20. The values shown were obtained considering the mean of 200 independent runs of the algorithms, a large enough number of runs to ensure the convergence of the values represented in Table 1. The efficiency factor (forth row) is obtained by division operation of the values from the third row by those from the second row, and is indicative of the computation time that can be saved by PFP-SubSim algorithm as compared to Algorithm 2. Observe that the results are fairly similar when ζ is high because reaching the final threshold is not in this case a rare event under the model given by Equation 20. However, lower values of the decay parameter ζ implies that reaching the threshold is less probable or even a rare event under the model, which is precisely the situation when PFP-SubSim exploits its higher efficiency. Indeed, note from Table 1 that about $^2/_3$ of the model evaluations can be saved by using PFP-SubSim algorithm as compared to Algorithm 2 for the lowest value of ζ, i.e. $\zeta = 5 \times 10^{-3}$ (fifth column), which implies an efficiency factor of 2,72. These results about PFP-SubSim efficiency will be further corroborated in the context of the next example of application.

4.2. Case study: Prediction of fatigue damage using monitoring data

In this section, the performance of the algorithm is investigated through a case study about damage prognostics in carbon fibre reinforced polymer (CFRP) laminates using structural health monitoring (SHM) data. In previous works by the authors [2, 11], a model-based prognostics methodology was presented which demonstrated efficiency for long-term predictions of fatigue damage in cross-ply laminates using SHM data. To avoid duplication of literature for this technique but conferring a sufficient conceptual framework, the relevant details from the referred methodology are presented here in a concise manner.

The starting point of this methodology involves the calculation of G, the energy released per unit crack area due to the formation of a new crack between two existing cracks as [42, 43]:

$$G = \frac{\sigma_x^2 h}{2\rho t_{90}} \left(\frac{1}{E_x^*(2\rho)} - \frac{1}{E_x^*(\rho)} \right) \tag{22}$$

where σ_x is the maximum applied axial tension, and h and t_{90} are the laminate and 90°-sublaminate half-thickness, respectively. The matrix micro-cracks density is denoted by $\rho = \frac{1}{2\bar{l}}$, where $\bar{l}$ is the half crack-spacing normalized by the 90° sub-laminate thickness. The term $E_x^*(\rho)$, as a function of ρ, is the effective Young's modulus due to the current damage state which can be modeled through micro-damage mechanics models [44]. In this case study, the *shear-lag* [45, 46] micro-damage model is adopted to be simpler and robust [47], such that:

$$E_x^* = \frac{E_{x,0}}{1 + a\frac{1}{2\bar{l}} R(\bar{l})} \tag{23}$$

where $E_{x,0}$ is the undamaged longitudinal Young's modulus of the overall laminate, and a is a function of the mechanical and geometrical properties of the laminate, which can be obtained using the classical laminate plate theory [48] (see also [49] for further details about the calculation of a). The term $R(\bar{l})$, known as the *average stress perturbation function*, is defined by [47]:

$$R(\bar{l}) = \frac{2}{\xi} \tanh(\xi \bar{l}) \tag{24}$$

where

$$\xi^2 = G_{yz} \left(\frac{1}{E_y} + \frac{t_{90}}{t_\phi E_x^{(\phi)}} \right) \tag{25}$$

In the last equation, G_{yz}, E_y, and $E_x^{(\phi)}$ are laminate mechanical parameters which are described next. The details about the geometry parameters involved in Equations 22 to 25, together with information about the pattern of micro-cracks, are given in Figure 5.

Next, the modified Paris' law [50] is used to model the evolution of matrix micro-cracks density as a function of fatigue cycle $n \in \mathbb{N}$, as follows:

$$\rho_n = \rho_{n-1} + A \left(\Delta G(\rho_{n-1}) \right)^\alpha \tag{26}$$

where A and α are fitting parameters. The term ΔG is the increment of energy release (recall Equation 22) evaluated for the maximum and minimum stress in the cyclic load series, i.e. $\Delta G = G|_{\sigma_{max}} - G|_{\sigma_{min}}$.

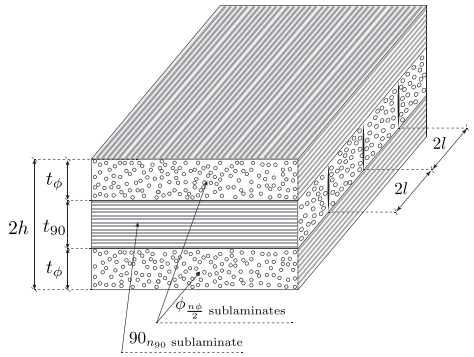

Figure 5: Illustration for microscopic damage pattern in $\left[\phi_{\frac{n_\phi}{2}}/90_{n_{90}}/\phi_{\frac{n_\phi}{2}}\right]$ laminate along with basic geometrical parameters.

4.2.1. Stochastic embedding

The progression of damage is modeled at every cycle n by focusing on the matrix-cracks density ρ_n and the normalized effective stiffness $D_n = {}^{E_x^*}/_{E_{x,0}}$, defining a joint state transition equation of two components as follows:

$$x_{1_n} = \rho_n = \underbrace{f_{1_n}(\rho_{n-1}, \theta)}_{\text{Eq. (26)}} + v_{1_n} \tag{27a}$$

$$x_{2_n} = D_n = \underbrace{f_{2_n}(\rho_n, \theta)}_{\text{Eq. (23)}} + v_{2_n} \tag{27b}$$

where $x_n = (x_{1_n}, x_{2_n}) \in \mathbb{R}^2$ is the actual system response at time n. Subscripts 1 and 2 correspond to the damage subsystems, namely, matrix-crack density and normalized effective stiffness, respectively. The vector $v_n = (v_{1_n}, v_{2_n}) \in \mathbb{R}^2$ corresponds to the model error vector of the overall system, which, by the Principle of Maximum Information Entropy (PMIE) [51–53], is chosen as a Gaussian, i.e. $v_n \sim \mathcal{N}\left(0, \left[\sigma_{v_{1_n}}, \sigma_{v_{2_n}}\right] I_2\right)$, where $\left[\sigma_{v_{1_n}}, \sigma_{v_{2_n}}\right]$ are the standard deviations of v_n and I_2 is the identity matrix of order 2, so they can be readily sampled.

Let us now consider that the vector $y_n = \left(y_{1_n}, y_{2_n}\right) = \left(\hat{\rho}_n, \hat{D}_n\right)$ represents the measurements of the system output x_n and also that a measurement function is added to the state-space model to account for the measurement error $w_n \in \mathbb{R}^2$:

$$y_{1_n} = \hat{\rho}_n = x_{1_n} + w_{1_n} \tag{28a}$$

$$y_{2_n} = \hat{D}_n = x_{2_n} + w_{2_n} \tag{28b}$$

where $\hat{\rho}_n, \hat{D}_n$ denote the measured matrix-cracks density and normalized effective stiffness, respectively. As stated before, the PMIE is used to choose w_n to be distributed as zero mean Gaussian PDF, i.e. $w_n \sim \mathcal{N}\left(0, \left[\sigma_{w_{1_n}}, \sigma_{w_{2_n}}\right] I_2\right)$, where $\left[\sigma_{w_{1_n}}, \sigma_{w_{2_n}}\right]$ are the standard deviations of w_n and I_2 the identity matrix of order 2. In Equations 27 and 28, the model parameters θ are selected among the complete set of mechanical

and geometrical parameters describing Equations 22 to 26 (see Table 2) through a Global Sensitivity Analysis based on variances [49, 54], resulting $\theta = \left(\alpha, E_x, E_y, t, \sigma_{v_{1_n}}, \sigma_{v_{2_n}}\right)$ as the vector of model parameters. The rest of parameters can be fixed at some point within their range of variation, (e.g. the mean value) without significantly influencing the output uncertainty.

4.2.2. Results

For the prediction of the PDF of EOL/RUL, SHM measurements of micro-cracks density and stiffness reduction from a fatigue test are considered for a 15.24×25.4 [cm] laminate coupon with dogbone geometry and $[0_2/90_4]_s$ stacking sequence. The mechanical properties along with their probabilistic information are listed in Table 2. In this example, $a = 0.1325$ (recall Equation 23), according to [49]. The test was conducted under load-controlled tension-tension fatigue loadings with a frequency of $f = 5$ [Hz], and maximum applied loads of 31.13 [KN], which represents an equivalent maximum stress of 80% of their ultimate stress. The ratio between the minimum and maximum applied stress per cycle was set to 0.14. Further details about this test can be found in [55] and also in the Composites dataset, NASA Ames Prognostics Data Repository [56] (damage data used in this example correspond to laminate L1S19). In [57], the details about the SHM methodology are provided. A summary of the dataset used for this study is extracted from [56] and provided here in Table 3.

At each prognostic step, which corresponds to each of the fatigue cycles when SHM data are available as shown in the first row of Table 3, Algorithm 1 is run by using $N = 5000$ particles and systematic importance resampling. Initial values for the damage states are $x_0 = (\rho_0, D_0)$, where $\rho_0 = 0.1$ [$^{\text{cracks}}/_{mm}$] and $D_0 = 1$ (dimensionless). The standard deviation of the measurement error parameters are set to $\sigma_{w_{1,n}} = 0.05$ [$^{\text{cracks}}/_{mm}$] and $\sigma_{w_{2,n}} = 0.01$, taking them as known. Chosen prior PDFs for model parameters $\theta = (\theta_1, \theta_2, \ldots \theta_6)$ are specified in Table 2. The diagonal elements of the covariance matrix W_n (recall Equation 6) are appropriately selected through initial test runs and set to 0.5% of the 5^{th}-95^{th} band of the prior PDFs for the j^{th} component of θ.

Updated damage states at time n are propagated forward in time by using PFP-SubSim algorithm. For illustration purposes, results of micro-cracks density and stiffness loss are shown in Figure 6 for prediction time $n = 3 \cdot 10^4$. Three simulation levels ($m = 3$) are employed in Figure 6 by using $P_0 = 0.5$ and $M = 2.4 \cdot 10^4$ samples per simulation level. The total amount of samples is thus $N_T = (2.4 + 1.2 + 1.2) \cdot 10^4$, because $MP_0 = 1.2 \cdot 10^4$ samples from each conditional level are used as seeds to start the next simulation level. In Figure 6, the triangles represent the time (in cycles) when matrix micro-cracks density will reach the final threshold $b = 418$ $[\#cracks \cdot m^{-1}]$, together with its associated threshold for stiffness 0.88, which are known from [56], and also shown in Table 3. These thresholds define a failure region given by $\bar{\mathcal{U}} = \{(\rho_n, D_n) \in \mathbb{R}^2 : 0 < \rho_n < 418,\, 0 < D_n < 0.88\}$.

In this example, $P_0 = 0.5$ is adopted to easily visualize how predicted samples are distributed among

different subsets, since the recommended near-optimal value of $P_0 = 0.2$ for Subset Simulation produced $m = 1$ conditional levels for the majority of times of prediction. The estimation of RUL by PFP-SubSim algorithm is plotted against time in Figure 7. The results shown in Figure 7 are satisfactory in the sense

Type	Parameter	Nominal value	Units	Prior PDF
Mechanical	E_x	$127.55 \cdot 10^9$	Pa	$\mathcal{LN}(\ln(127.55 \cdot 10^9), 0.1)$
	E_y	$8.41 \cdot 10^9$	Pa	$\mathcal{LN}(\ln(8.41 \cdot 10^9), 0.1)$
	G_{xy}	$6.20 \cdot 10^9$	Pa	Not applicable
	G_{yz}	$2.82 \cdot 10^9$	Pa	Not applicable
	ν_{xy}	0.31	–	Not applicable
Geometrical	t	$1.5 \cdot 10^{-4}$	m	$\mathcal{LN}(\ln(1.5 \cdot 10^{-4}), 0.1)$
Fitting	α	1.80	–	$\mathcal{LN}(\ln(1.80), 0.2)$
	A	$1 \cdot 10^{-4}$	–	Not applicable
Errors	σ_{v_1}	–	$\frac{\#\text{ cracks}}{m \cdot \text{cycle}}$	$\mathcal{U}(0.5, 1.5)$
	σ_{v_2}	–	–	$\mathcal{U}(0.001, 0.003)$

Table 2: Prior information and nominal values of main parameters used in calculations.

Fatigue cycles	10^1	10^2	10^3	10^4	$2 \cdot 10^4$	$3 \cdot 10^4$	$4 \cdot 10^4$	$5 \cdot 10^4$	$6 \cdot 10^4$	$7 \cdot 10^4$	$8 \cdot 10^4$	$9 \cdot 10^4$
ρ [# cracks/m]	98	111	117	208	270	305	355	396	402	402	407	418
D	0.954	0.939	0.930	0.924	0.902	0.899	0.888	0.881	0.896	0.872	0.877	0.885

Table 3: Experimental sequence of damage for cross-ply $[0_2/90_4]_s$ CFRP laminate. The data are presented for micro-cracks density (ρ) and normalized effective stiffness (D).

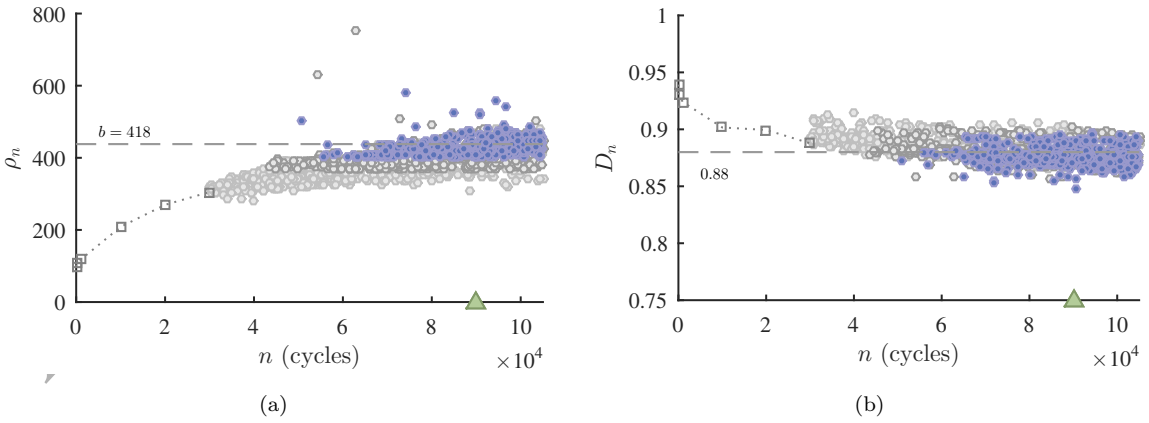

Figure 6: PFP-SubSim output for predicting (a) matrix micro-cracks density and (b) normalized effective stiffness at $n = 3 \cdot 10^4$. Each subset is represented by samples (circles) in the $\mathcal{Z}$ space, where the latest intermediate predictive samples are marked in dark purple circles.

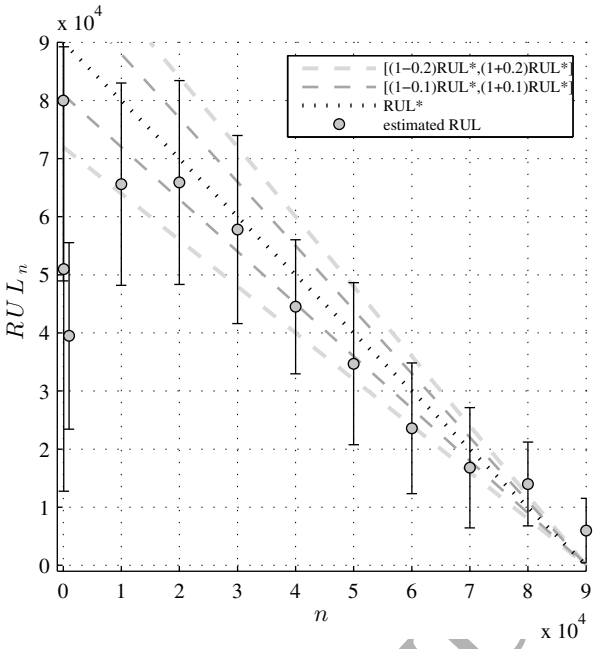

Figure 7: Results for RUL predictions together with their quantified uncertainty by the interquartile bars.

that the proposed algorithm has the ability to obtain the RUL with high precision, except in the first stage of the fatigue process, which corresponds to the interval of cycles required for the data to train the model parameters θ. To help evaluating the prediction accuracy and precision, two shaded cones of accuracy are used at 10% and 20% of the true RUL, denoted here as RUL*, following the methodology given by [58]. Observe also from Figure 7 that accuracy seems to depart from true RUL at the final stage, in the sense that the estimated mean values for the RUL (labeled by the gray circles) leave progressively the accuracy area as fatigue cycles evolve from $n = 5 \cdot 10^4$. Such behavior has been previously reported in [2, 11] and was found to be related with the asymptotic behavior of the damage progression in composites.

4.2.3. Discussion

In this section, comparative exercises have been carried out to investigate the computational improvement and accuracy that can be achieved using PFP-SubSim algorithm with respect to a traditional prognostics algorithm, like Algorithm 2. It should be noted that the results shown here are extensible to other applications different than composite materials, since they are focused on efficiency aspects about the simulation engines of the competing algorithms.

The first exercise of interest regards the evaluation of the probability of prospective failure during a prescribed interval of future instants $(n, n+\ell] \subset \mathbb{N}$, as follows:

$$P(\bar{\mathcal{U}}) = \int_{\bar{\mathcal{U}}} p(z_{n:n+\ell}|y_{0:n}) dz_{n:n+\ell} \approx \frac{1}{N_T} \sum_{k=1}^{N_T} \mathbb{I}_{\bar{\mathcal{U}}}(z_{n:n+\ell}^{(k)}) \tag{29}$$

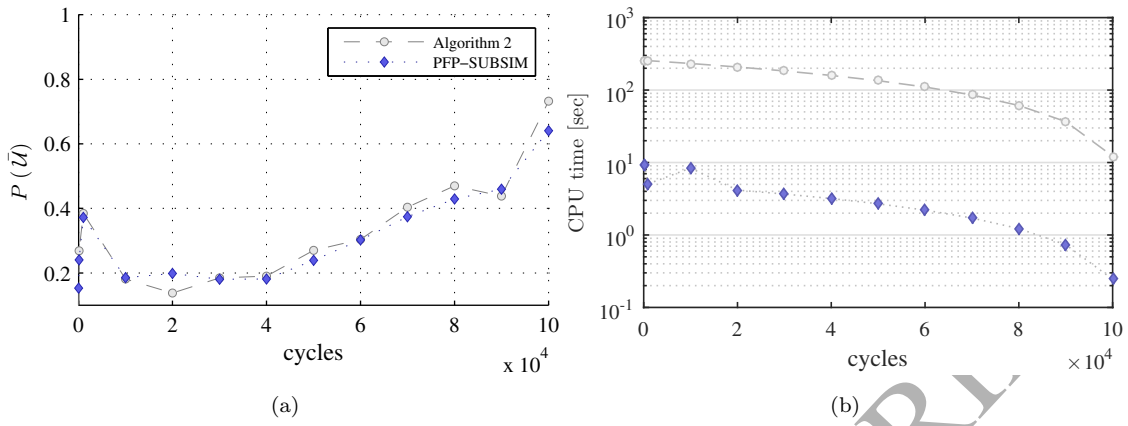

Figure 8: (a) Plots for $P(\bar{\mathcal{U}})$ estimation using PFP-SubSim algorithm in comparison to Algorithm 2 used as benchmark. (b) CPU time employed by each algorithm, represented in logarithmic scale and expressed in seconds.

where $\left\{z_{n:n+\ell}^{(k)}\right\}_{k=1}^{N_T}$ are N_T predictive samples simulated according to the PDF $p(z_{n:n+\ell}|y_{0:n})$. Note that the approximation given by Equation 29 is equivalent to the standard Monte Carlo method for evaluating probability integrals as a mathematical expectation. Note also that this method has a well-known drawback in cases of small values for $P(\bar{\mathcal{U}})$, by the fact that a huge number of simulations are required to achieve acceptable estimation accuracy, which may increase the computational cost significantly. By PFP-SubSim, $P(\bar{\mathcal{U}})$ can be straightforwardly evaluated via the the conditional probabilities involved in Subset Simulation (recall Equation 17):

$$P(\bar{\mathcal{U}}) = P(\bar{\mathcal{U}}_1)\prod_{j=2}^{m} P(\bar{\mathcal{U}}_j|\bar{\mathcal{U}}_{j-1}) \approx P_0^m \tag{30}$$

Figure 8a shows estimations of the probability of prospective failure $P(\bar{\mathcal{U}})$ obtained at different prediction instants n using comparatively the simulation engines of PFP-SubSim algorithm and Algorithm 2. To avoid an excessive computational cost for this exercise, the simulations are restricted to lie within the interval $(n, 10^5] \subset \mathbb{N}$, since such interval is enough to highlight the differences between both algorithms. The CPU times required by each algorithm are plotted against the prediction instants in Figure 8b, which have been obtained using a 3.5 GHz double-core system. The results shown for Algorithm 2 were obtained using a large enough amount of samples for the estimation of $P(\bar{\mathcal{U}})$ to be sufficiently accurate, hence used here as benchmark values. Observe that the estimations of $P(\bar{\mathcal{U}})$ produced by PFP-SubSim algorithm agree well with the benchmark values given by Algorithm 2, even when PFP-SubSim requires significantly less CPU time. Note also that the predicted probability values are high in comparison to $P_0 = 0.2$, hence a significant improvement in efficiency is expected for PFP-SubSim algorithm if the comparison is performed for failure events of lower probability values. To this end, an additional comparative exercise has been carried out by using the unit coefficient of variation (c.o.v.) $\triangle$ as a quality indicator of the $P(\bar{\mathcal{U}})$ estimate. The c.o.v. is used

since it gives a measure of efficiency that is inherent to the algorithm because it is invariant to the number of simulated samples[3] [25]. This is of special interest for our exercise since the total amount of samples required by PFP-SubSim algorithm varies depending on the failure probability value to be estimated. The results are reported in Figure 9 for both algorithms in terms of the unit c.o.v. versus different threshold levels, which correspond to decreasingly lower values for $P(\bar{\mathcal{U}})$ estimates (labeled by the text box). In this exercise, the $P(\bar{\mathcal{U}})$ estimations are evaluated at cycle $n = 8 \cdot 10^4$. To quantitatively assess the statistical properties for the prognostics estimates, the numerical values of each plot from Figure 9 are obtained by considering the mean of 200 independent runs of both algorithms. Observe that when PPF-SubSim algorithm is employed, the unit c.o.v. values increase slowly towards the region of thresholds that entail lower probability values (higher threshold values for matrix micro-cracks density), whilst for Algorithm 2 the unit c.o.v. grows steeply in that region. However, the performance of both algorithms is fairly similar for higher probability values, (say of the order of $P_0 = 0.2$), which confirms that PFP-SubSim gets its highest efficiency when predictions based on simulating rare events are required. Further discussion can be provided by comparatively examining PFP-SubSim with other sampling-based prognostics methods for uncertainty propagation [34, 59], or by exploring methodologies to enhance Subset Simulation efficiency [60]. However they are considered out of the scope of this work.

[3] Any stochastic algorithm for estimating $P_{\bar{\mathcal{U}}}$ has a c.o.v. of the form c.o.v. $= \triangle/\sqrt{N_T}$

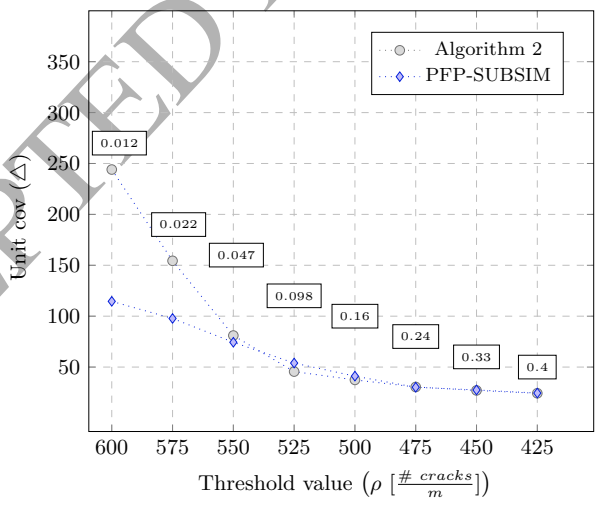

Figure 9: Results for the unit c.o.v. of $P(\bar{\mathcal{U}})$ obtained using PFP-SubSim algorithm as compared to that obtained using Algorithm 2. The corresponding failure probability is labeled by a text box.

5. Conclusions

This paper presented a novel prognostics algorithm, the PFP-SubSim algorithm, that combines the prognostics principles with the Subset Simulation method. The algorithm gets its efficiency by adaptively drawing samples over a nested sequence of intermediate failure regions (subsets), until a predefined final region is reached. These regions were defined in an adaptive manner, which avoids preliminary calibrations. An example of application along with a case study were provided to illustrate the computational efficiency that can be gained with PFP-SubSim. The results indicated that our algorithm is highly efficient for the prognostics of processes involving rare-event simulations, whilst its behavior is fairly similar to a standard prognostics algorithm when probabilities are not so small. For that reason, PFP-SubSim can be considered as a general purpose prognostics algorithm, which is specially suited for rare-event prediction. Further research is needed to investigate the extension of the proposed algorithmic framework for the case of multidimensional state spaces (say $n_z > 10$). In particular, an improvement of much interest would be the application of adaptive methods to optimally scale the artificial evolution of the model parameters in multidimensional state spaces.

Acknowledgment

Dr. Manuel Chiachío is a Research Fellow of the Lloyd's Register Foundation (LRF), a charitable foundation in the UK helping to protect the life and property by supporting engineering-related education, public engagement, and the application of research. The two first authors would like to specially thank the Prognostics Center of Excellence at NASA Ames, which kindly hosted them during part the course of this work. Authors would also like to thank the Structures and Composites Lab (under auspices of Prof. Fu-Kuo Chang) at Stanford University for experimental data, and NASA ARMD/AvSafe project SSAT, which provided partial support for this work. Finally, they would also like to thank Prof. James L. Beck from California Institute of Technology and Dr. Abhinav Saxena from General Electric Global Research, for their valuable guidance through Subset Simulation and prognostics algorithms, respectively.

References

[1] J. Chiachío, M. Chiachío, A. Saxena, K. Goebel, Prognostics design for structural health management, in: Emerging Design Solutions in Structural Health Monitoring Systems, IGI Global, 2015, pp. 234–273.

[2] M. Chiachío, J. Chiachío, A. Saxena, K. Goebel, An energy-based prognostic framework to predict evolution of damage in composite materials, in: Structural Health Monitoring (SHM) in Aerospace Structures, Woodhead Publishing-Elsevier, 2016, pp. 447–477.

[3] R. Reinertsen, Residual life of technical systems; diagnosis, prediction and life extension, Reliability Engineering and System Safety 54 (1) (1996) 23–34.

[4] K. Javed, R. Gouriveau, N. Zerhouni, State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels, Mechanical Systems and Signal Processing 94 (2017) 214–236.

[5] E. Zio, Reliability engineering: Old problems and new challenges, Reliability Engineering and System Safety 94 (1) (2009) 125–141.

[6] J. D. Andrews, T. R. Moss, Reliability and risk assessment, Professional Engineering Publishing Limited, 2002.

[7] D. An, N. H. Kim, J.-H. Choi, Practical options for selecting data-driven or physics-based prognostics algorithms with reviews, Reliability Engineering & System Safety 133 (2015) 223–236.

[8] S. Sankararaman, K. Goebel, Uncertainty in prognostics and systems health management, International Journal of Prognostics and Health Management 6 (010).

[9] P. Baraldi, F. Mangili, E. Zio, Investigation of uncertainty treatment capability of model-based and data-driven prognostic methods using simulated data, Reliability Engineering & System Safety 112 (2013) 94–108.

[10] F. Cadini, E. Zio, Model-based Monte Carlo state estimation for condition-based component replacement, Reliability Engineering and System Safety 94 (1) (2009) 752–758.

[11] J. Chiachío, M. Chiachío, S. Shankararaman, A. Saxena, K. Goebel, Condition-based prediction of time-dependent reliability in composites, Reliability Engineering and System Safety 142 (2015) 134–147.

[12] B. Saha, J. R. Celaya, P. F. Wysocki, K. F. Goebel, Towards prognostics for electronics components, in: Aerospace conference, 2009 IEEE, IEEE, 2009, pp. 1–7.

[13] M. Daigle, S. Kulkarni, Electrochemistry-based battery modeling for prognostics, in: Proceedings of the Annual Conference of the Prognostics and Health Management Society, 2013, Vol. 1, 2013, pp. 249–261.

[14] M. Jouin, R. Gouriveau, D. Hissel, M.-C. Péra, N. Zerhouni, Degradations analysis and aging modeling for health assessment and prognostics of pemfc, Reliability Engineering & System Safety 148 (2016) 78–95.

[15] B. Saha, K. Goebel, J. Christophersen, Comparison of prognostic algorithms for estimating remaining useful life of batteries, Transactions of the Institute of Measurement and Control 31 (3-4) (2009) 293–308.

[16] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, Signal Processing, IEEE Transactions on 50 (2) (2002) 174–188.

[17] A. Doucet, N. De Freitas, N. Gordon, An introduction to sequential Monte Carlo methods, in: A. Doucet, N. De Freitas, N. Gordon (Eds.), Sequential Monte Carlo methods in practice, Springer, 2001, pp. 3–14.

[18] E. Zio, G. Peloni, Particle filtering prognostic estimation of the remaining useful life of nonlinear components, Reliability Engineering and System Safety 96 (3) (2011) 403–409.

[19] M. Orchard, G. Kacprzynski, K. Goebel, B. Saha, G. Vachtsevanos, Advances in uncertainty representation and management for particle filtering applied to prognostics, in: International Conference on Prognostics and Health Management (PHM, 2008), IEEE, 2008, pp. 1–6.

[20] M. Daigle, K. Goebel, Multiple damage progression paths in model-based prognostics, in: Aerospace Conference, 2011 IEEE, IEEE, 2011, pp. 1–10.

[21] M. Daigle, K. Goebel, Model-based prognostics with fixed-lag particle filters, in: Proceedings of the Annual Conference of the Prognostics and Health Management Society, 2009, Vol. 1, 2009, pp. 249–261.

[22] J. R. Celaya, A. Saxena, K. Goebel, Uncertainty representation and interpretation in model-based prognostics algorithms based on Kalman filter estimation, in: Annual Conference of Prognostics and Health Management Society, PHM Society, 2012, pp. 12–24.

[23] M. Corbetta, S. Sbarufatti, M. Giglio, Optimal tuning of particle filtering random noise for monotonic degradation processes, in: Proceedings of the third European Conference of the Prognostics and Health Management Society, IEEE, 2016, pp. 479–489.

[24] S. Au, J. Beck, Estimation of small failure probabilities in high dimensions by Subset Simulation, Probabilistic Engineering Mechanics 16 (4) (2001) 263–277.

[25] S. Au, J. Ching, J. Beck, Application of Subset Simulation methods to reliability benchmark problems, Structural Safety 29 (3) (2007) 183–193.

[26] J. Ching, S. Au, J. Beck, Reliability estimation of dynamical systems subject to stochastic excitation using Subset Simulation with splitting, Computer Methods in Applied Mechanics and Engineering 194 (12-16) (2005) 1557–1579.

[27] M. Chiachío, J. Chiachío, A. Saxena, K. Goebel, An efficient simulation framework for prognostics of asymptotic processes-a case study in composite materials., in: Proceedings of the European Conference of the Prognostics and Health Management Society, Nantes, France 2014, PHM Society, 2014, pp. 202–214.

[28] E. Myötyri, U. Pulkkinen, K. Simola, Application of stochastic filtering for lifetime prediction, Reliability Engineering and System Safety 91 (2) (2006) 200–208.

[29] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE-Proceedings-F 140 (1993) 107–113.

[30] J. Liu, M. West, Combined parameter and state estimation in simulation-based filtering, in: A. Doucet, N. Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Statistics for Engineering and Information Science, Springer New York, 2001, pp. 197–223.

[31] M. J. Daigle, K. Goebel, Model-based prognostics with concurrent damage progression processes, Systems, Man, and Cybernetics: Systems, IEEE Transactions on 43 (3) (2013) 535–546.

[32] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and computing 10 (3) (2000) 197–208.

[33] J. Handschin, D. Q. Mayne, Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering, International Journal of Control 9 (5) (1969) 547–559.

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

[35] J. Ching, J. L. Beck, K. A. Porter, R. Shaikhutdinov, Bayesian state estimation method for nonlinear systems and its application to recorded seismic response, Journal of Engineering Mechanics 132 (4) (2006) 396–410.

[36] G. Kitagawa, Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, Journal of computational and graphical statistics 5 (1) (1996) 1–25.

[37] A. Kong, J. S. Liu, W. H. Wong, Sequential imputations and Bayesian missing data problems, Journal of the American statistical association 89 (425) (1994) 278–288.

[38] F. Liang, C. Liu, J. Chuanhai, Advanced Markov chain Monte Carlo methods, Wiley Online Library, 2010.

[39] K. Zuev, J. Beck, S. Au, L. Katafygiotis, Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions, Computers & Structures 93 (2011) 283–296.

[40] M. Chiachío, J. L. Beck, J. Chiachío, G. Rus, Approximate Bayesian computation by Subset Simulation, SIAM Journal on Scientific Computing 36 (3) (2014) A1339–A1358.

[41] S. Au, J. Beck, Subset Simulation and its application to seismic risk based on dynamic analysis, Journal of Engineering Mechanics 129 (8) (2003) 901–917.

[42] J. A. Nairn, The strain energy release rate of composite microcracking: a variational approach, Journal of Composite Materials 23 (11) (1989) 1106–1129.

[43] J. A. Nairn, Some new variational mechanics results on composite microcracking, in: Proc. 10th International Conference on Composite Materials (ICCM-10) Whistler BC, Canada, 1995.

[44] R. Talreja, C. V. Singh, Damage and failure of composite materials, Cambridge University Press, 2012.

[45] K. Garrett, J. Bailey, Multiple transverse fracture in 90° cross-ply laminates of a glass fibre-reinforced polyester, Journal of Materials Science 12 (1) (1977) 157–168.

[46] A. Highsmith, K. Reifsnider, Stiffness-reduction mechanisms in composite laminates, Damage in composite materials, ASTM STP 775 (1982) 103–117.

[47] R. Joffe, J. Varna, Analytical modeling of stiffness reduction in symmetric and balanced laminates due to cracks in 90 layers, Composites Science and Technology 59 (11) (1999) 1641–1652.

[48] J. Reddy, Mechanics of Laminated Composite Plates and Shells, Theory and Analysis, 2nd. Edition, CRC Press, 2004.

[49] J. Chiachío, M. Chiachío, A. Saxena, S. Shankararaman, K. Goebel, Bayesian model selection and parameter estimation for fatigue damage progression models in composites, International Journal of Fatigue 70 (2015) 361–373.

[50] J. Nairn, S. Hu, The initiation and growth of delaminations induced by matrix microcracks in laminated composites, International Journal of Fracture 57 (1) (1992) 1–24.

[51] E. Jaynes, Information theory and statistical mechanics, Physical Review 106 (4) (1957) 620–630.

[52] E. Jaynes, Probability theory: the logic of science, Ed. Bretthorst, Cambridge University Press, 2003.

[53] J. Beck, Bayesian system identification based on probability logic, Structural Control and Health Monitoring 17 (7) (2010) 825–847.

[54] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, Wiley-Interscience, 2008.

[55] A. Saxena, K. Goebel, C. Larrosa, V. Janapati, S. Roy, F. Chang, Accelerated aging experiments for prognostics of damage growth in composites materials, in: The 8th International Workshop on Structural Health Monitoring, FK. Chang, (Ed), Stanford, CA, 13-15 September 2011, Vol. 1, 2011, pp. 1283–1291.

[56] A. Saxena, K. Goebel, C. Larrosa, F. Chang, CFRP Composites dataset, NASA Ames Prognostics Data Repository, [http://ti.arc.nasa.gov/project/prognostic-data-repository], NASA Ames, Moffett Field, CA (2008).

[57] C. W. Larrosa, F. K. Chang, Monitoring fatigue-induced transverse matrix cracks in laminated composites using built-in acousto-ultrasonic techniques, Structural Health Monitoring 15 (3) (2016) 335–350.

[58] A. Saxena, J. Celaya, B. Saha, B. Saha, K. Goebel, Metrics for offline evaluation of prognostic performance, International Journal of the PHM Society 1 (1) (2010) 4–23.

[59] F. Cadini, C. Sbarufatti, M. Corbetta, M. Giglio, A particle filter-based model selection algorithm for fatigue damage identification on aeronautical structures, Structural Control and Health Monitoring (2017) in press.

[60] V. Papadopoulos, D. G. Giovanis, N. D. Lagaros, M. Papadrakakis, Accelerated subset simulation with neural networks for reliability analysis, Computer Methods in Applied Mechanics and Engineering 223 (2012) 70–80.