A fast Monte Carlo method for model-based prognostics based on stochastic calculus

Matteo Corbetta* Chetan Kulkarni †
SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035

This work proposes a fast Monte Carlo method to solve differential equations utilized in model-based prognostics. The methodology is derived from the theory of stochastic calculus, and the goal of such a method is to speed up the computation of the probability density functions describing the evolution of the quantity of interest over time. In the prediction case studies proposed in this paper, the stochastic differential equations describe quantities of interest directly or indirectly related to the degradation of a monitored system. The method allows the estimation of the probability density functions by solving the deterministic equation and approximating the stochastic integrals using samples of the model noise. By so doing, the prognostic problem is solved without the Monte Carlo simulation based on Euler's forward method, which is typically the most time consuming task of the prediction stage. Three different prognostic scenarios are presented as a proof of concept: (i) life prediction of electrolytic capacitors, (ii) remaining time to discharge of Lithium-ion batteries, and (iii) prognostic of cracked structures under fatigue loading. The paper shows how the method produces probability density functions that are statistically indistinguishable from the distributions estimated with Euler's forward Monte Carlo simulation. However, the proposed solution is orders of magnitude faster when computing the time-to-failure distribution of the monitored system. The approach may enable complex real-time prognostics and health management solutions with limited computing power.

Nomenclature

α, β	=	capacitance loss model parameters
$\beta', \gamma', \lambda, \mu, \nu_L$	=	battery model parameters
γ	=	fatigue crack growth model parameter
$\boldsymbol{\theta}$	=	model parameter vector
ν	=	significance level of t-test
ξ	=	standard Gaussian random variable
σ	=	standard deviation function
ω	=	model noise
B	=	Brownian motion or Wiener process
C_l	=	capacitance loss
C'	=	fatigue crack growth model parameter
C, m	=	Paris' law model parameters
E	=	battery's delivered energy
F	=	crack shape function
L	=	applied far-field stress
N	=	number of Monte Carlo samples
P	=	required power
R	=	battery internal resistance
S	=	state-of-charge
T_F	=	time-to-failure
V	=	voltage
a	=	semi-crack length

*Research Engineer, matteo.corbetta@nasa.gov, AIAA member.
†Research Engineer, chetan.s.kulkarni@nasa.gov, Senior AIAA member.

$f_{\boldsymbol{\theta}}$	=	model equation function parameterized by $\boldsymbol{\theta}$
k	=	discrete time variable
n	=	fatigue load cycles
p, q	=	general probability density functions
t	=	continuous time variable
$t_{\nu,N}$	=	t-test acceptance value with confidence level ν and degrees of freedom N
$\tilde{t}$	=	t-test result
u	=	input variable
x	=	general system's state variable
z	=	standard Gaussian random variable

I. Introduction

Monte Carlo (MC) techniques are widely employed in the field of uncertainty management, providing a direct method for simulation and integration [1]. They are utilized in a number of scientific fields, from physics, to finance, to systems engineering, to operational research. In model-based prognostics, which can be defined as the science of making predictions specifically relying on physics-based or empirical models [2], a large number of MC algorithms have been presented.

One of the main advantages of MC methods for model-based prognostics is their ability to propagate uncertainty through nonlinear functions and in time domain, particularly when compared to other techniques that are limited by the nonlinearity of the system or non-Gaussian distributions, e.g., Kalman filters [3]. Examples of MC methods applied in prognostics include the works in [4, 5], which utilized a modification of Markov chain MC algorithm for degradation prognosis. He, Williard, Osterman, and Pecht [6] used MC sampling to predict the state of health of Lithium-ion batteries, while Orchard and Vachtsevanos utilized a sequential MC algorithm to perform diagnosis and prognosis of a structural component [7]. A number of sequential MC-based solutions to prognostics have been proposed since then [8–11], making sequential MC, also called particle filter or sequential importance sampling, a very popular algorithm for model-based prognosis [12]. All the cited works, and many others, use MC to propagate samples through the equations describing the system's evolution, predict potential failures, and the corresponding time to failure. The sample propagation through the system's evolution equation is performed using Euler's forward method, and it will be called, for the sake of brevity, Monte Carlo simulation (MCS) henceforth. This prediction step based on MCS requires high computing power because of the many samples necessary to represent the probability density functions (pdfs) of the quantities of interest correctly, and because of the Euler's integration time step size dt that should be small enough to prevent instability of the solution. Such computational costs may be prohibitive in many real-time prognostic applications, especially in case of constrained computing platforms. Real-time onboard health monitoring of small unmanned aerial vehicles, space rovers or space robots are typical example of engineering systems where the computing hardware should be minimized to maximize payload and/or maneuverability.

For this reason, this paper proposes to look at MC sampling methods for model-based prognostics from the perspective of stochastic calculus, to reduce the required computation time and, therefore, the necessary computing power. The work analyzes the relationship between the state-space formulation typically adopted in prognostic problems and the form of some fundamental stochastic differential equations (SDEs). Under certain assumption, the methodology can speed up the estimation of the system's state pdf many steps ahead in the future, avoiding the usage of MCS to simulate the entire system's dynamics in a probabilistic fashion. By so doing, the prediction of the system's state over time may be completed in a fraction of the time required by MCSs.

To prove the advantages of the stochastic calculus-based approach, linear and nonlinear case studies are presented to cover different prognostics and health management scenarios; (i) aging electrolytic capacitors, (ii) state of charge of lithium-ion batteries, (iii) damaged structures subject to fatigue loading. The results show that the features characterizing the uncertainty of the system's dynamics calculated with the proposed method are preserved when compared with estimations from MCSs. The approximated pdfs are statistically equivalent to the ones obtained from MCSs in terms of estimated moments and relative entropy. However, the computational time required by the proposed approach is drastically lower in most of the analyzed cases, especially when computing the time-to-failure (TTF) pdfs.

The remainder of the paper is organized as follows: Section II summarizes the prognostic problem and gives more intuitions behind the formulation of ordinary differential equations utilized in prognostics and the high computational costs of MC-based solutions. Section III defines the basic properties of model noise and SDEs that can be utilized to represent the uncertainty of the system's state variable(s). Section IV presents the case studies where the proposed

Table 1 simplified architecture of model-based prognostic algorithm based on MCS. Input l represents the number of steps to be predicted.

Input: $x_k^{(i)} \sim p(X_k),\ i = 1, \ldots, N;\ l$
Output: $x_{k+l}^{(i)} \sim p(X_{k+l}), \quad i = 1, \ldots, N$
for *each* $x_k^{(i)} \sim p(X_k)$ **do**
  for *each* $\tau \in \{1, \ldots, l\}$ **do**
    $\omega_{k+\tau-1}^{(i)} \sim p(\Omega_{k+\tau-1})$
    $x_{k+\tau}^{(i)} = x_{k+\tau-1}^{(i)} + f_{\boldsymbol{\theta}}(x_{k+\tau-1}^{(i)}, u_{k+\tau-1}, \omega_{k+\tau-1}^{(i)})\mathrm{d}t_{k+\tau-1}$
  end
end

methodology has been applied, while Section V draws some conclusions on the performed work and its future expansions.

II. Summary of model-based prediction

This section describes some key aspects of model-based prognostics necessary to estimate the system's state variable pdfs ahead into the future, i.e., to make a prediction of the system's state according to the underlying model. For the sake of simplicity, the system's state is defined by a scalar variable x.

Most of the literature on model-based prognostics, being the models empirically derived or physics-based, utilizes a state-space formulation, which composes of a model equation and an observation equation. The model equation is often defined as the ordinary differential equation in Eq. (1). The discretization of the equation using Euler's forward method, visible in the second row of Eq. (1), enables the numerical solution of the model:

$$\begin{aligned} \dot{x} &= f_{\boldsymbol{\theta}}(x, u, \omega), \\ x_k &= x_{k-1} + f_{\boldsymbol{\theta}}(x_{k-1}, u_{k-1}, \omega_{k-1})\Delta t_k . \end{aligned} \tag{1}$$

The intuition behind Euler's forward method is that, given a known initial condition, the numerically-computed discrete increments of the state variable approximate the continuous rate of change for $\Delta t_k \to 0$. A simple architecture of a MC-based prognostic algorithm utilizing (1) is shown in table 1, where the pseudo-code grounds on the following, simplifying assumptions:

1) the system's state at the current time step k is described by the pdf $p(X_k)$, whose approximation is available,
2) the model parameters collected into the vector $\boldsymbol{\theta}$ are known, deterministic, and constant,
3) the input u is deterministic and known, and
4) the model noise is a random variable described by $p(\Omega)$, which may be time-varying, and samples $\omega^{(i)}$ are independent and identically distributed.

It should be noted that many model-based prognostic methods use direct or indirect observations Y of the quantities of interest to update the system's state in a Bayesian fashion, therefore producing a conditional pdf, $p(X_k|Y_{0:k})$. In those cases, the resulting prediction pdf at time step $k+l$ would be $p(X_{k+l}|Y_{0:k})$. Since this paper does not discuss observability, further discussion on the availability of measures y is omitted. Some of the hypotheses (1-4) may be relaxed without losing the validity of the proposed approach. However, this varies on a case-basis and further analyses may be required.

The outer loop of the algorithm in table 1, in most cases, can be easily optimized by using a vectorization procedure, which allows the propagation of all (or multiple) samples in one line of code, without the need of a *for* loop. However, the inner loop in table 1 is still solved using a MCS in time-domain, where all samples are propagated through the model equation from time step k up to the required time step l. This work aims at accelerating such a inner loop taking advantage of the properties of stochastic calculus framework.

The estimation of the remaining useful life pdf, which is a fundamental task in model-based prognostics, consumes most of the computational power required by most prognostic algorithms. The remaining life prediction is carried out by pre-defining a threshold of the system's state variable, x_{th}, which should not be exceeded to guarantee operational and functional requirements of the monitored system. The approximation of the remaining useful life pdf is generated by estimating when the system's state reaches the threshold, thus computing the TTF distribution, and then subtracting the current time. In order to do so, all the samples $x^{(i)}$ drawn during the simulation have to reach x_{th}. However, since MCS

relies or random realization of the model noise, some of those samples may be subject to a sensibly slower dynamic when compared with the deterministic (i.e., average) solution, thus requiring a high number of discrete time steps to reach the threshold and so increasing the overall computation time necessary to estimate the TTF distribution. Details on the prediction process and, more in general, on prognostic and health management algorithms, are available in [2].

The goal of this work translates, in practice, in a method that allows drawing N samples from the distribution of the system's state variable many steps ahead in the future, or from the TTF distribution, without relying on MCS that utilizes Euler's forward method.

III. Using stochastic calculus properties in model-based prediction

The similarity between the state-space formulation utilized in prognostic and SDEs can be easily understood by considering a state-space model with additive noise and the general formulation of SDEs:

$$\begin{aligned} \dot{x}_t &= f_{\boldsymbol{\theta}}(x_t, u_t) + \omega_t \ , \\ x_k &= x_{k-1} + f_{\boldsymbol{\theta}}\left(x_{k-1}, u_{k-1}\right) \Delta t_k + \omega_{k-1} \Delta t_k \ , \end{aligned} \tag{2}$$

$$\dot{X}_t = f_{\boldsymbol{\theta}}(X_t, U_t) + \sigma(t, X_t)\, \xi_t \ , \tag{3a}$$

$$X_t = X_0 + \int_0^t f_{\boldsymbol{\theta}}(X_s, U_s)\, \mathrm{d}s + \int_0^t \sigma(s, X_s)\, \mathrm{d}B_s \ , \tag{3b}$$

$$X_k \approx X_0 + \sum_{s=0}^{k-1} f_{\boldsymbol{\theta}}(X_s, U_s)\, \Delta t_s + \sum_{s=0}^{k-1} \sigma(s, X_s)\, \Delta B_s \ . \tag{3c}$$

Equations (2) are easily derived from Eq. (1), while Eqs. (3) are the formulation (Eq. (3a)) and closed form solutions (Eqs. (3b)-(3c)) of a SDE with model noise $\sigma(t, X_t)\, \xi_t$. The random variable ξ_t is typically embedded in the stochastic term $\sigma(t, X_t)$, but here has been explicitly extracted from the σ function for reasons that will be clearer later in the section. This formulation is similar to the one adopted in [13], Section 4, and [14]. Equation (3c) shows approximation of the solution using Riemann's sum. It should be noticed that Eq. (2) has been written assuming, as initial condition, the system's state at the previous time step, while SDEs are typically written using X_0 as initial condition. Moreover, with an abuse of notation, many works on model-based prognostic confuses random variables and their realizations by describing them with lower case letters (see Eq. (2)), while typical SDE notation uses capital letters for random variables. In this section, the two different notations have been preserved.

First, let us consider the SDE in Eq. (3) and the simple case where the random perturbation is a time-invariant, standard normal random variable, i.e., $\xi_t = z \sim \mathcal{N}(0, 1)$ and $\sigma(t, X_t) = \sigma$. Under those assumptions, the term σ becomes the standard deviation of the model noise. When integrating (3a), the random perturbation is multiplied by the time increment $\mathrm{d}t$ thus generating a Wiener process or Brownian motion [14], and therefore the integration of the perturbation term can be approximated as in Eq. (4).

$$\int_0^t \sigma\, z\, \mathrm{d}t = \int_0^t \sigma\, \mathrm{d}B = \sum_{s=0}^{k-1} \sigma \Delta B_s \tag{4}$$

If we assume the model noise in Eq. (2) to be $\omega_t \sim \mathcal{N}(0, \sigma^2)$, then $\omega = \sigma\, z$, where $z \sim \mathcal{N}(0, 1)$. The model noise term in Eq. (2) therefore coincides with an element of the Riemann sum utilized in Eq. (4):

$$\omega_{k-1}\, \Delta t_k = \sigma\, z_{k-1}\, \Delta t_k = \sigma\, \Delta B_k \ . \tag{5}$$

This simple equivalence allows us to solve the model equation utilizing the properties of Brownian motion and more in general the properties derived in the SDE domain. Some properties of Brownian motion useful to solve the prediction problem are listed in Eqs. (6).

$$\mathrm{d}B \sim \mathcal{N}(0, \mathrm{d}t) \tag{6a}$$

$$B_{t_2} - B_{t_1} \sim \mathcal{N}(0, t_2 - t_1) \tag{6b}$$

Equation (6a) means that we can generate samples of $\mathrm{d}B$ as $\mathrm{d}B^{(i)} = \sqrt{\mathrm{d}t}\, z^{(i)}$, where $z^{(i)}$ is the i-th sample from an independent Normal standard distribution. Similarly to Eq. (6a), Eq. (6b) shows that the difference between two realizations of Brownian motion at time t_1 and t_2, with $t_1 < t_2$, is described by a Gaussian distribution with variance $t_2 - t_1$.

The discussion presented above does not require the model noise variance to be stationary, but the model noise should remain additive, i.e., the σ function should not depend on the system's state X, $\sigma \neq \sigma(X)$. Nevertheless, the case studies presented hereafter assume a constant model noise variance, since it can cover a variety of model-based prognostic applications, as illustrated in Section IV.

IV. Applications

This section presents three prognostic case studies where the model equations are solved using two methods: traditional MCS and the proposed solution, which will be called SDE henceforth, since it relies on works carried out in the field of SDEs. Both techniques are used to: (i) estimate the pdf of the system's state variable at a certain time step T and (ii) estimate the pdf of TTF, T_F, defined as the time when the system's state variable(s) exceeds a predefined threshold.

Since different models requires different solutions of the SDE, a generalized framework is not provided. Rather, the three case studies refer to different complexity of the prediction problem: a linear model with additive Gaussian noise, a nonlinear model with additive Gaussian noise, and a nonlinear model with multiplicative, non-Gaussian noise.

The pdfs calculated with the two methods are collated to one another for each case study. The Kullback-Leibler (KL) divergence [15] is utilized to measure the difference between pdfs, Eq. (7). Being MCS the standard method to perform model-based prognostic, the distribution calculated via MCS has been considered here the reference distribution.

$$\mathrm{KL}(p||q) = \int_{-\infty}^{+\infty} p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x \approx - \sum_{i=0}^{N_*-1} p_i \log \frac{q_i}{p_i} \tag{7}$$

The discrete formulation has been taken from [16]. The probability densities p and q in Eq. (7) refer to MCS and SDE, respectively, and the probability density values p_i and q_i are calculated by dividing the range of independent variable x into 400 elements, $x^{(i)}, \forall i = 1, \ldots, N_*$, $N_* = 400$, equally spaced from the minimum $x_{\min}$ to the maximum $x_{\max}$ values obtained by the two methods, i.e., $x_{\min} = \min\{\min(x_{\mathrm{MCS}}), \min(x_{\mathrm{SDE}})\}$, $x_{\max} = \max\{\max(x_{\mathrm{MCS}}), \max(x_{\mathrm{SDE}})\}$.

In addition to KL divergence, a *t-test* statistics has been performed to confirm the equality of the mean of the two populations. The t-test validates evidence for difference in means between the two sets. Here, the variance of the two distributions have been assumed equal but unknown. The t-test hypothesis becomes:

$$\begin{cases} H_0 : & \mu_{\mathrm{MCS}} = \mu_{\mathrm{SDE}} \\ H_1 : & \mu_{\mathrm{MCS}} \neq \mu_{\mathrm{SDE}} \; ; \quad \nu = 0.05 \, , \end{cases} \tag{8}$$

where ν is the significance level, kept constant for all case studies, while $\mu_{\cdot}$ indicate the MC mean of the population. A visual comparison between the distributions has also been carried out using histograms and non-parametric kernel density estimators with same properties to minimize external factors that may influence the graphs: number of bins = 20 and Gaussian kernel with bandwidth = 0.5.

The time required to compute the pdfs has been recorded and reported here to emphasize the speed of the two methods. The absolute time value is not relevant as it would change according to the machine utilized for the computation. However, the relative difference is a clear indication of whether one of the two methods is faster than the other. All three case studies have been simulated using Python 2.7 [17] and the figures were generated with the *matplotlib* package [18]. Other libraries utilized for this work are *SciPy* and *NumPy*, [19]. All simulations were executed by first fixing the random seed of NumPy to 1. With an abuse of notation, random variables and their realizations are represented by the same letter (we do not distinguish between capital or lower case), using the superscript (i) to indicate realizations. The simulation parameters utilized to produce the results change according to the case study and they are reported in the related subsections.

A. Case study 1: prognostic of electrolytic capacitors

This case study utilizes the model proposed in [20] to predict the end of life of electrolytic capacitors. The degradation model describes the capacitance loss in percentage terms. Its differential form with additive Gaussian

model noise ω follows Eq. (9), [20].

$$\dot{C}_l = \alpha C_l - \alpha\beta + \omega \tag{9}$$

MCS requires a series of samples of the capacitance loss, starting from an initial state $C_l(0) = C_{l,0}$, and propagate such samples through the discrete version of Eq. (9) until the pre-defined time T or until the threshold $C_{l,th}$ defining the end of life of the capacitor. Equation (9) is a non-homogeneous SDE, where α and β are model parameters, C_l is the system's state variable, and ω is the random perturbation term. The solution of a non-homogeneous SDE is the summation of (i) the homogeneous solution, (ii) the periodic solution, and (iii) the integration of the random perturbation term over time.

The homogeneous solution of Eq. (9) is:

$$C_{l,h} = Ae^{\alpha\, t}\ , \tag{10}$$

while the periodic solution is:

$$C_{l,p} = \beta\ . \tag{11}$$

By using deterministic initial conditions $C_l(0) = C_{l,0} = 0$, the integral form of the SDE solution follows:

$$C_l(t) = e^{\alpha t}\left(-\beta + \beta e^{-\alpha t} + \int_0^t \sigma e^{-\alpha s}\mathrm{d}B_s\right)\ . \tag{12}$$

As clear from Section III, the integral in Eq. (12) is a stochastic integral with respect to Brownian motion. It can be approximated by a finite sum of samples following Eq. (13):

$$\int_0^T \sigma e^{-\alpha s}\mathrm{d}B_s \approx \sum_{s=0}^{k-1} \sigma e^{-\alpha t_s}\Delta B_s^{(i)}\ , \forall i = 1, \ldots, N\ , \tag{13}$$

where the superscript (i) refers to the i-th sample and the subscript s refers to the time step. The variable k is the total number of time steps, defined as $k = T\,\Delta t^{-1}$. The same, intuitive rule of Euler's forward method applies to the stochastic integral approximation: using large Δt would cause imprecise approximations. Contrariwise, using small Δt, the computation approaches an invariant distribution as $k \to \infty$ and $N \to \infty$.

Figure 1 shows the simulation of the capacitance loss over time. The deterministic curve have been calculated by superimposing the homogeneous and the periodic solution, neglecting the noise term. In this specific case, since the analytical solution of the ordinary differential equation is available, the black curve in Fig. 1 can also be generated using Eq. (12) by setting $\sigma = 0$. MCS provides the swarm of light blue curves, while the SDE method does not require to compute the capacitance loss at every time step, but only at the desired time step(s). To compare the pdfs computed with the two methods, the SDE-based solution has been calculated at $t_* = \{50, 100, 150, 200\}$ [h] (red dots).

Figure 2 shows the pdfs and corresponding histograms at different time steps. The location of the bins have not been fixed, so we let the *matplotlib* function *hist* [18] decide where to place the bin centers, according to the location of the samples. Even with this degree of freedom from the histogram function, the bins appear extremely close to one another.

Table 2 shows the first two moments of the distributions, while Table 3 compares the pdfs by showing the KL divergence, and the t-test statistics on the means of the two sample sets. The KL divergence is of the order of $1e-3$, which indicates very similar distributions. The t-test results show how the null hypothesis H_0 cannot be rejected in any case, with all $\tilde{t}$ values below $t_{\nu/2,2N-2}$. The t-test showed a higher $\tilde{t}$ at time 200 h, and one may wonder whether longer prognostic horizons would increase the difference between the distributions calculated with the two method. However, a further run of the two algorithms by changing the random seed generated the following: $\tilde{t} = \{0.497, 0.527, 0.556, 0.494\}$ for the four time steps in t_*. Several other runs using different random seeds generated volatile $\tilde{t}$, but all of them below $t_{\nu,2N-2}$. Besides the numerical analysis in Tables 2 and 3, the reader may notice the slight discrepancy between the pdfs shown in Fig. 2, particular in Figs. 2a, 2c, and 2d, since the mode of the distributions appears at different values of the capacitance loss. Such an inconsistency of the mode value is caused by the sampling error. Figure 3 shows the same distributions computed with a very large number of samples ($N = 5000$, against 1000 samples used in Figure 2). In this latter case the mode, as well as the overall shape of the distributions, are almost indistinguishable.

The time necessary to compute the distributions is shown in the last column of Table 3. The SDE method did not provide any advantage in the presented case, since the time required to compute the capacitance loss pdf was higher than the time required by the MCS approach to accomplish the same task. Instead, it appears that the proposed approach is slower, and multiple runs of the algorithms produce similar outcomes. The reasons for this result lie in Eqs. (12) and (13). The numerical approximation of the stochastic integral requires a large number of random samples from the

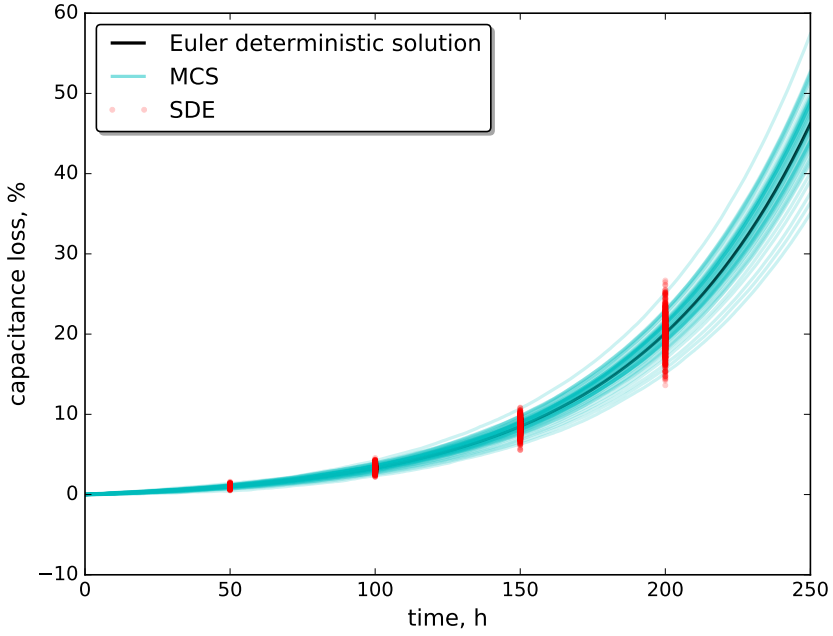

Fig. 1 Capacitance loss over time simulated using model (9). Only a subset of the MCS solution is shown in the figure to appreciate the different trajectories.

Table 2 Moments of the distributions.

t [h]	$\mathbb{E}[C_l]$		$\mathbb{V}[C_l]$	
	MCS	SDE	MCS	SDE
50	1.01928	1.01715	0.0239	0.02484
100	3.31516	3.32281	0.14912	0.1520
150	8.48215	8.48590	0.77970	0.7794
200	20.08643	20.19213	3.96765	3.96166

model noise distribution, neglecting the advantage of having a closed form solution for $C_l(t)$. However, the SDE method becomes much faster than MCS when estimating the TTF distribution, as presented below.

In the TTF problem, the capacitance loss threshold was arbitrary set to 8%, therefore, the TTF pdf can be approximated when all samples from the MCS reach the value 0.08. When using the SDE method, instead, the TTF pdf can be computed without propagating the samples through the model equation. The expected value of the time-to-failure $\mathbb{E}[T_F]$ can be easily calculated by inverting Eq. (**??**) and setting $C_l = C_{l,th} = 0.08$, Eq. (14).

$$\mathbb{E}[\mathrm{T_F}] = \frac{1}{\alpha} \ln\left(1 - \frac{C_{l,th}}{\beta}\right). \tag{14}$$

Once the expected value of T_F has been calculated, the i-th sample of the TTF pdf can be extracted by inverting Eq. (12):

$$\mathrm{T_F}^{(i)} = \frac{1}{\alpha} \ln \frac{C_{l,th} - \beta}{\int_0^{\mathbb{E}[T_f]} \sigma e^{-\alpha s} \mathrm{d}B_s - \beta} \qquad \forall\, i = 1, \ldots, N\,, \tag{15}$$

where $T_F^{(i)}$ is the i-th sample of the TTF distribution, and the stochastic integral is solved similarly to Eq. (13).

The pdf estimation of the samples calculated with the two methods is shown in Fig. 4. The similarity between the two distributions is reported in Table 4 in terms, again, of KL divergence and t-test statistics, and the computing time is

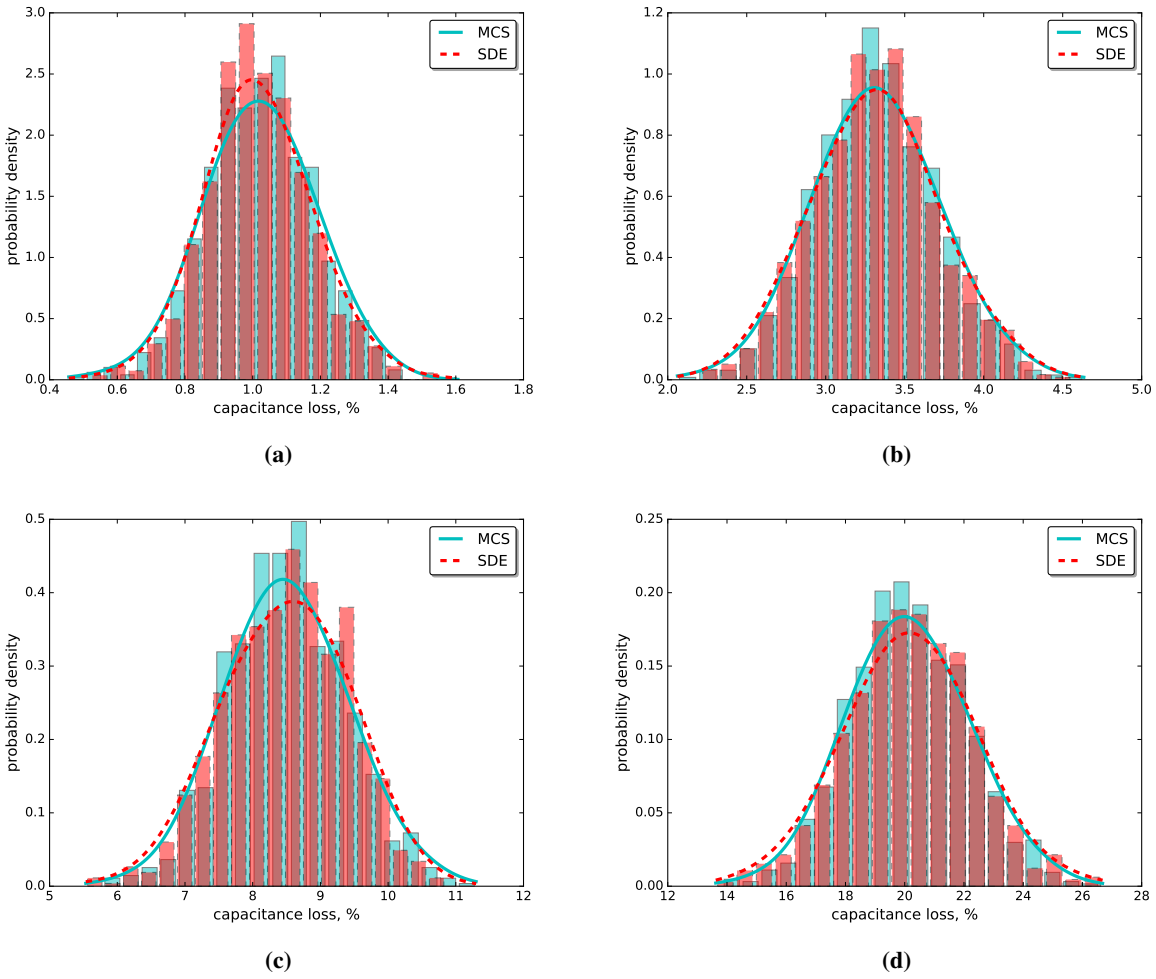

Fig. 2 Capacitance loss distributions calculated at t=50 [h] (a), t=100 [h] (b), t=150 [h] (c), and t=200 [h] (d). The histograms have been normalized to match the height of the pdfs.

reported in the last two columns. The computing time of the SDE method is hundreds of times faster than MCS. The reason for this large discrepancy lies in the step-by-step procedure that MCS uses to propagate the samples. Samples $T_F^{(i)}$ do not require any recursive function to be computed.

This application showed how the pdfs of the quantity of interest over time can be computed by taking advantage of the properties of stochastic calculus. According to the results presented here, there is no evidence of statistical difference between the distributions computed with MCS and the proposed method.

The numerical values of model parameters and random variables utilized to produce the results are the following:

$$
\begin{aligned}
N &= 1000, \text{ [-]} \\
\boldsymbol{\theta} &= \{\alpha, \beta\} = \{0.0162, -0.82\}, \text{ [-, -]} \\
\Delta t &= 0.01, \text{ [h]} \\
C_{l,0} &= 0, \text{ [\%]} \\
\sigma^2 &= 2\text{e} - 2, \text{ to simulate the capacitance loss over time} \\
\sigma^2 &= 2\text{e} - 3, \text{ to predict the time-to-failure}
\end{aligned}
$$

The choice to use two different model noise variances for the prediction of the $C_l(t)$ distribution and the TTF

distribution was dictated by the dispersion of the samples close to the lower bound ($C_l \approx 0$). A large variance of the model noise causes some of the samples $C_l^{(i)}$ to become negative, thus wasting many time steps to propagate unlikely samples. If σ^2 increases up to a certain level, the computation may not converge because some samples will never reach the threshold. This problem is related to the appropriateness of additive Gaussian noise in monotonic degradation processes rather than the approach presented in this paper, and has been recently discussed in literature [21]. Small values of the variance are therefore needed to complete the simulation. A more appropriate noise model would allow the algorithm designer to keep the variance of the two predictions, i.e., $C_l(t)$ pdf and TTF pdf, equal. Since this problem is out of the scope of this paper, it is not discussed any further and the reader may refer to [21] for additional insights about model noise strategies for model-based prognostics.

B. Case study 2: predicting the remaining time to discharge of Lithium-ion batteries

The second case study aims at predicting the state-of-charge (SOC) of Lithium-ion batteries based on a simplified empirical model presented in [22]. The model was proposed to deal with constrained computing platform like small unmanned aerial vehicles. The model equation is composed of three system's state variables: internal resistance R, delivered energy E, and state-of-charge S. The model was described by the authors in its discrete form, although it has been rewritten here in its continuous formulation, Eq. (16).

$$\begin{aligned} \dot{R} &= \omega_R \\ \dot{S} &= -\frac{P}{E} + \omega_S \\ \dot{E} &= \omega_E \end{aligned} \tag{16}$$

Power, P, is input to the model and assumed deterministic, known, and constant, i.e., $P(t) = P$. This model equation is a simple nonlinear model, where the nonlinearity is introduced by the term E^{-1} in the SOC equation. The complete state-space model includes Eq. (16) and the nonlinear measurement equation composed of open-circuit voltage v_{oc}, discharge current i, and battery voltage V, Eq. (17).

$$\begin{aligned} v_{oc} &= v_L + \lambda\, e^{\gamma' S} - \mu\, e^{-\beta' \sqrt{S}} \\ i &= \frac{v_{oc} - \sqrt{v_{oc}^2 - 4\,R\,P}}{2\,R} \\ V &= v_{oc} - i\,R + \omega_V \end{aligned} \tag{17}$$

Samples of voltage V can be computed by feeding samples of R and S to Eq. (17). The SDE method may give computational advantage considering that the three random variables of the model equation are generated using additive Gaussian noise (where, in particular, ω_R and ω_E are added to a null deterministic function). Similarly to Case Study 1, samples of the model noise ω can be generated by assuming a certain standard deviation σ and generating a sample from the standard normal distribution, z. This leads the Gaussian pdfs of the internal resistance and delivered energy at time t to assume the form:

$$\begin{aligned} R_t &\sim \mathcal{N}(R_0, \sigma_R^2\, t)\,, \\ E_t &\sim \mathcal{N}(E_0, \sigma_E^2\, t)\,. \end{aligned} \tag{18}$$

Table 3 Comparison between the pdfs computed using MCS and the propsed SDE method.

t [h]	$\mathrm{KL}(p_{\mathrm{MCS}} \| \| p_{\mathrm{SDE}})$	$\tilde{t}$	$t_{\nu/2,2N-2}$	computing time [s]	
				MCS	SDE
50	0.00097	0.305	1.961	0.22963	0.243095
100	0.00223	0.441		0.426782	0.468167
150	0.00143	0.095		0.627436	0.699553
200	0.00208	1.187		0.818853	0.989845

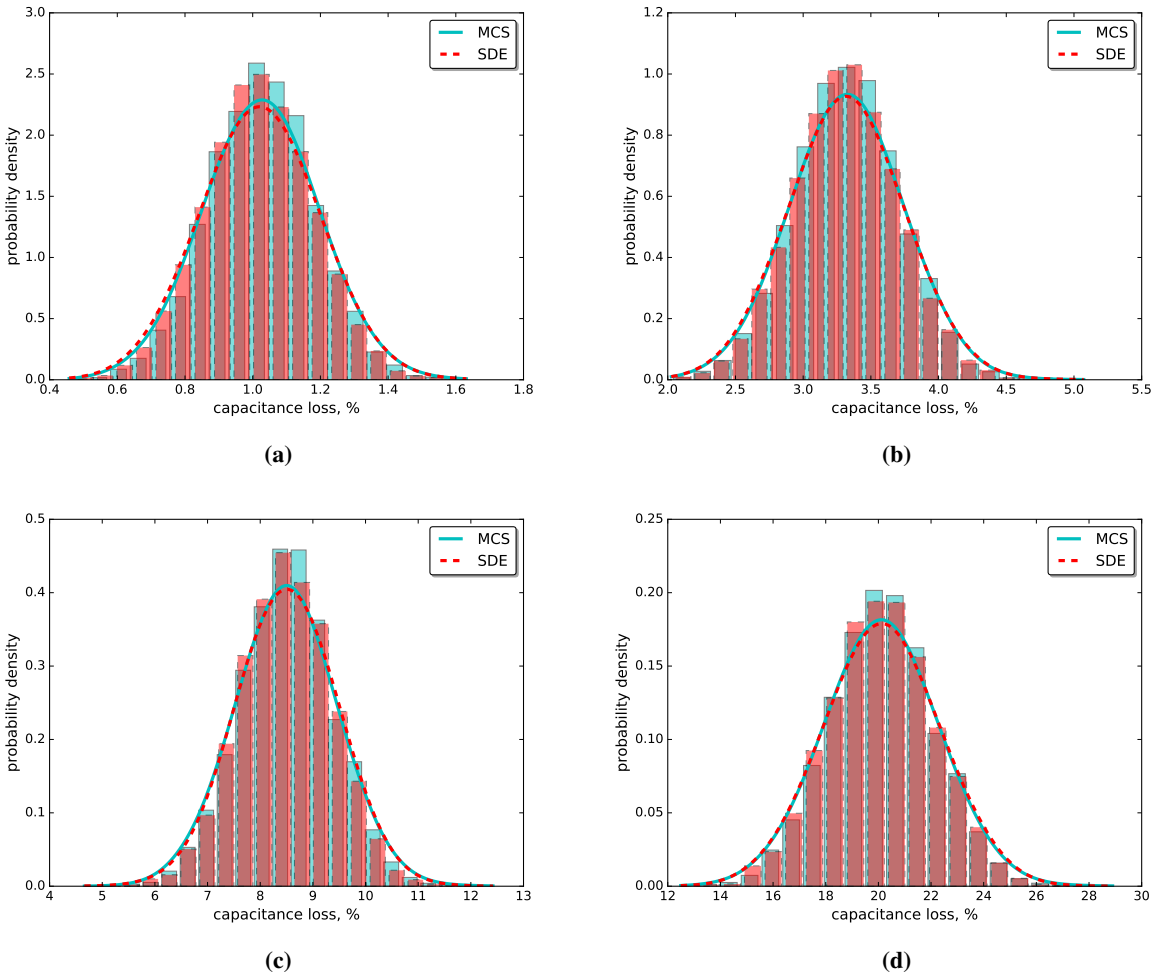

Fig. 3 Capacitance loss distributions calculated at t=50 [h] (a), t=100 [h] (b), t=150 [h] (c), and t=200 [h] (d) using N = 5000.

The variances σ_R^2 and σ_E^2 are the variances of the model noises ω_R and ω_E, respectively. In practice, Eq. (18) shows the well-known formula to extract samples from a random walk with initial values R_0 and E_0, at any time t. Being the samples of the delivered energy available from Eq. (18), the i-th sample of the SOC at time t follows:

$$S_t^{(i)} \sim \mathcal{N}\left(S_0 - \frac{P}{E_t^{(i)}}t, \sigma_S^2 t\right) . \tag{19}$$

The resulting samples of the state variables can then be utilized to estimate voltage of the battery and perform a prediction, as suggested in [22]. Similarly to the approach utilized in Case Study 1, the analysis of the distributions obtained from the two methods is discussed below.

Figure 5 shows the samples of the system's state variable in time domain. The light blue thick lines are samples generated through MCS, the red dots are samples generated with the proposed method, and the thick black line is the deterministic value of the system's state variables generated with Euler's forward method (often hidden behind the MCS samples). For the sake of brevity, only a single calculation using SDE method has been compared with MCS here. The figure shows how the samples generated with the proposed method well-align with the MCS samples, with no need to sample anything in between 0 and t = 200 s. The SOC pdf generated with the two methods is shown in Fig. 6. The distribution moments and the metrics utilized to evaluate the similarity of the two distributions are reported in Tables 5 and 6. The KL divergence strongly indicates similarities between the two pdfs and there is not enough evidence to define

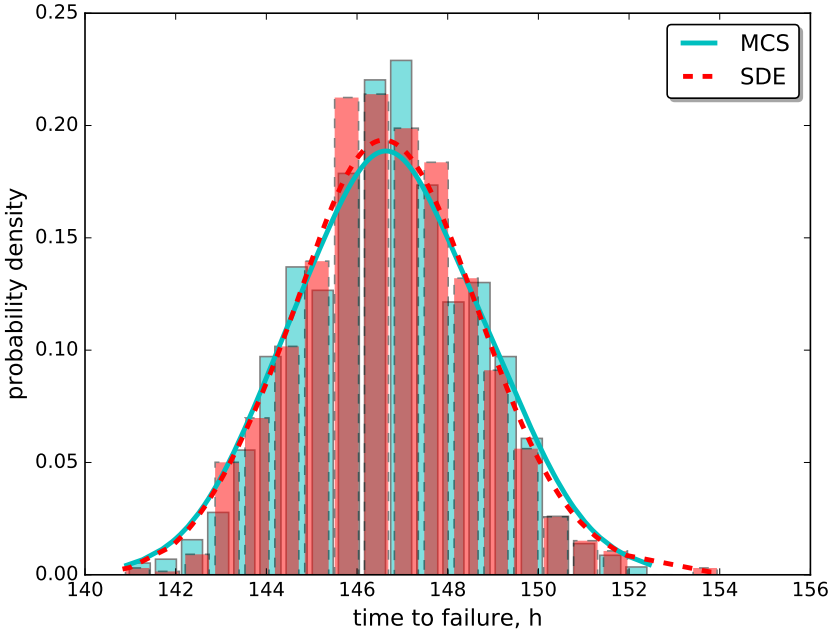

Fig. 4 Time-to-failure pdf computed by MCS and the proposed method, setting the capacitance loss threshold to 8 %. The histograms have been normalized to match the height of the pdfs.

Table 4 Comparison between time-to-failure distributions of capacitance loss C_l.

$\mathrm{KL}(p_{\mathrm{MCS}}\|\|p_{\mathrm{SDE}})$	$\tilde{t}$	$t_{\nu/2,2\,N-2}$	computing time [s]	
			MCS	SDE
0.001513	0.049	1.961	777.803	0.612

the mean values of the two pdfs different (with $\nu = 0.05$). This also implies the equivalence of the two voltage pdfs, which are shown in Fig 7. A quantitative analysis of the voltage pdfs including KL divergence, t-test, and estimation of first two moments is not reported here, but the results agree with the findings obtained for the SOC distributions. The computing time of the two methods is instead extremely different. The SDE method is again hundreds of times faster than MCS, thanks to Eqs. (18) and (19).

The prediction of the TTF distribution is usually constructed by defining a voltage threshold, V_{th}, below which the battery can not operate safely. Therefore, the objective is to directly sample the time instant when the voltage reaches such a threshold. On the other hand, the high nonlinearity of the relationship between V, R and S makes the simplification of the sampling problem using stochastic calculus hard to solve. It would require the inversion of (16) and (17), or any other method, to estimate the time instant when S and R generate the desired $V = V_{th}$. The authors decided to leave this task, if possible, to future works. The simulation and model parameters utilized to generate the

Table 5 Moments of the distributions calculated at T=200 s.

variable	$\mathbb{E}[\cdot]$		$\mathbb{V}[\cdot]$	
	MCS	SDE	MCS	SDE
E	202426.808	202427.237	686.576	709.896
R	0.02699	0.02699	4.7e-7	4.7e-7
S	0.985	0.985	4.6e-7	4.3e-7

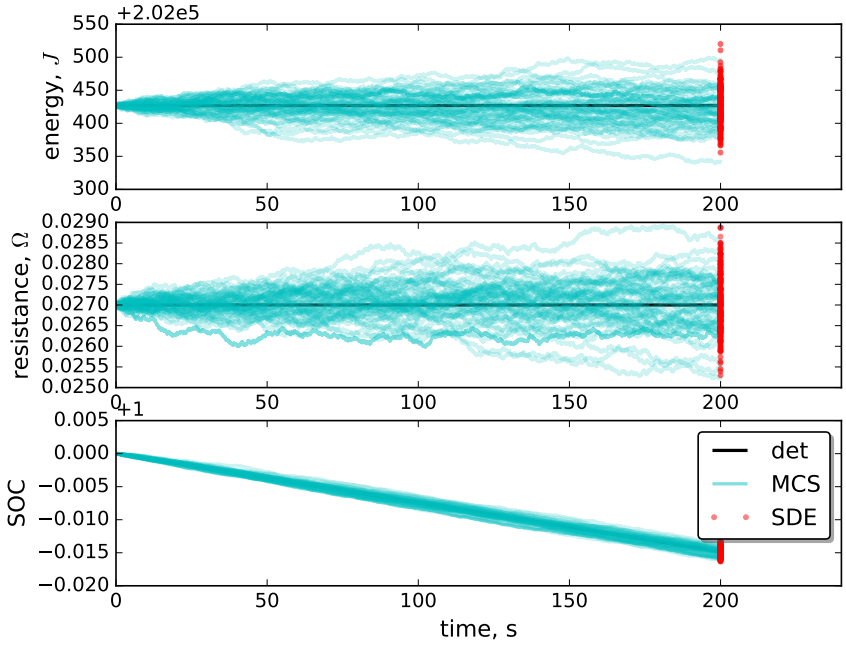

Fig. 5 Evolution of delivered energy (top), internal resistance (middle), and SOC (bottom) up to 200 s of simulation. Only a subset of the MCS solution is shown in the figure to appreciate the different trajectories.

Table 6 Comparing the estimated SOC distributions.

$\mathrm{KL}(p_{\mathrm{MCS}}\|\|p_{\mathrm{SDE}})$	$\tilde{t}$	$t_{\nu/2,2N-2}$	computing time [s]	
			MCS	SDE
0.00138	1.454	1.961	2.978	0.009

proposed results are reported below. The battery parameters come from the original paper [22] were the SOC model was extracted from. The last term is the variance of the voltage noise ω_V introduced in the voltage equation (17).

$$
\begin{aligned}
N &= 2500, \text{ [-]} \\
\Delta t &= 0.02, \text{ [s]} \\
\boldsymbol{u} &= u = P = 15, \text{ [W]} \\
\boldsymbol{\theta} &= \{\beta', \gamma', \lambda, \mu, v_L\} = \{8.482, 3.355, 0.046, 2.759, 11.148\}, [-] \\
\boldsymbol{x}_0 &= [R_0, S_0, E_0] = [0.027, 1.0, 202426.858], [\Omega, -, \mathrm{J}] \\
\sigma_E^2 &= 176.3, [\mathrm{J}^2] \\
\sigma_R^2 &= 12\mathrm{e}-7, [\Omega^2] \\
\sigma_S^2 &= 1.163\mathrm{e}-7, \text{ [-]} \\
\sigma_V^2 &= 0.01, [\mathrm{V}^2]
\end{aligned}
$$

C. Case study 3: fatigue damage prognosis of cracked structure

The third, last case study involve a well-known nonlinear model for fatigue crack growth (FCG) prediction. The model is based on the Paris' law equation [23], and this specific work utilizes some simplifying hypotheses:

1) the input is a constant-amplitude and constant-frequency fatigue loading, so a deterministic and direct relationship between elapsed time and number of applied load cycles exists,

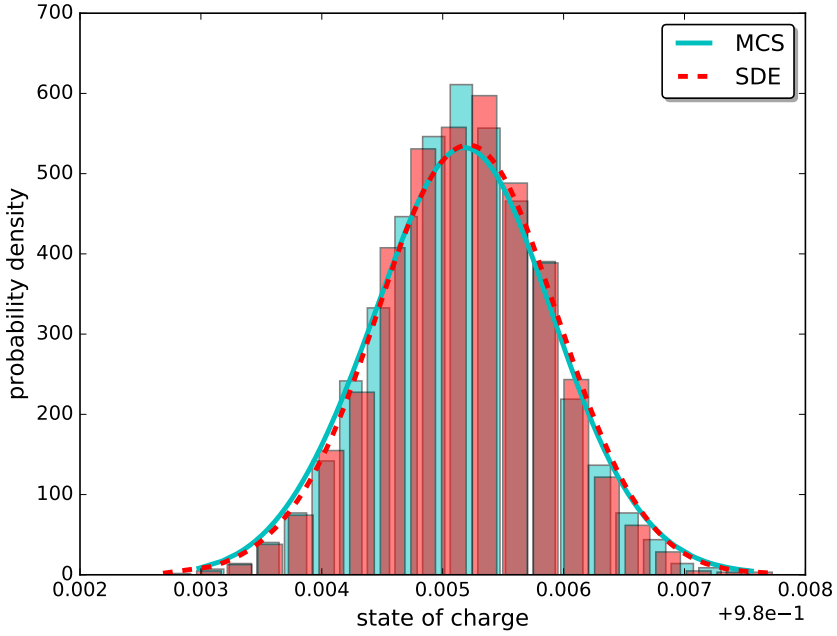

Fig. 6 SOC distributions at time t =200 s. The histograms have been normalized to match the height of the pdfs.

2) an initial crack size estimation is available, and
3) the crack nucleates and propagates in a simple structure where the crack shape function F can be assumed constant.

Not all the assumptions above are strictly necessary to implement the method. However, the authors left relaxing those hypotheses to future works. Moreover, many scientific papers on prognostic of FCG are also based on the same or similar simplifications [24], and the assumption of constant-amplitude fatigue loads may be considered valid for some mechanical systems, like rotating machinery. Therefore, the example proposed here is not different to other scenarios already discussed in literature.

The model is shown in Eq. (20). The typical Paris' law parameters utilized in mechanical engineering have been slightly reformulated. The model parameter C' collects the parameter C of the Paris' law, the crack shape function F, the stress amplitude $\Delta L = L_{\max} - L_{\min}$, and the constant $\sqrt{\pi}$. The model parameter γ is half of the Paris' law exponent, $\gamma = m/2$. The model noise is a multiplicative log-Normal random variable, with specific relationship between its mean and variance, in order to produce an unbiased propagation of the samples in time domain [21]. According to Paris' law and the model noise in [21], the semi-crack length dynamic follows:

$$\frac{\mathrm{d}a}{\mathrm{d}n} = C'\, a^{\gamma}\, e^{\omega}\,, \tag{20}$$

where a is the semi-crack length, expressed in mm*, and n is the load cycle variable. The load cycle domain substitutes the time domain here, since we assumed constant-amplitude and constant-frequency fatigue loading. In the deterministic case, the Fourier method (also known as method of separation of variables) solves the FCG prediction by separating the independent variable a and load cycle variable n, and integrating both sides, one side in da and the other side in dn. The constant arising from the integration can be set to 0. The Fourier method applies also to the stochastic model in Eq. (20), since the model noise e^{ω} has been assumed independent from the state variable. Equation (21) shows the integral form of the model:

*the crack length could also be expressed in meters or any other unit of length by a proper adjustment of Paris' law parameters and the equation of the *stress intensity factor*, [25]

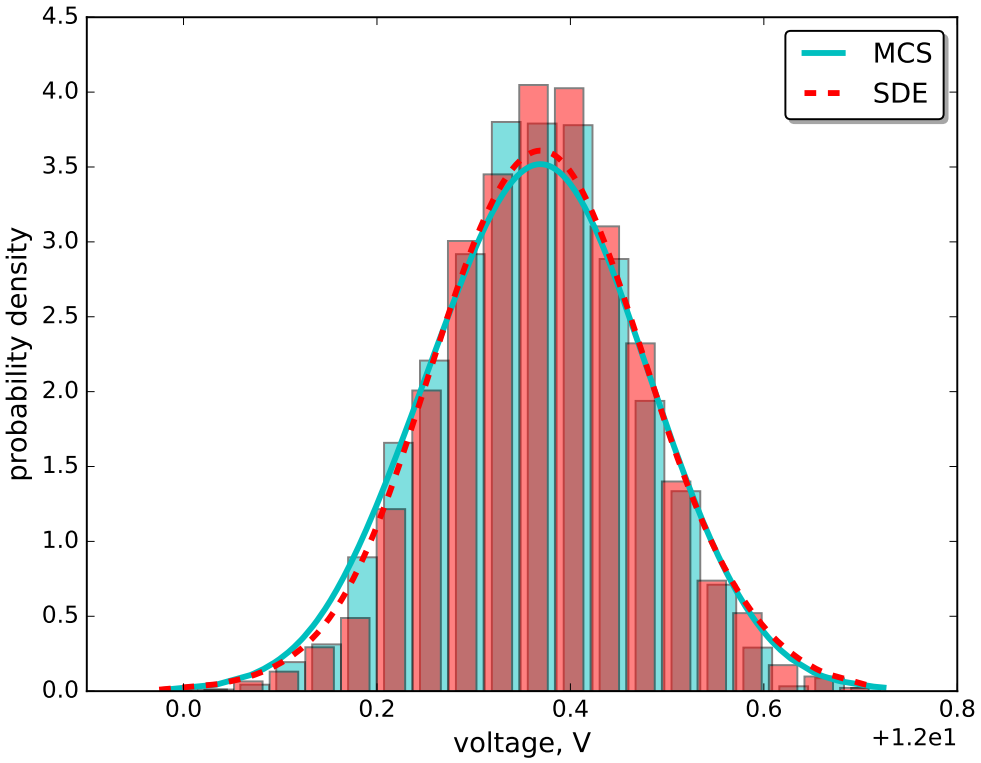

Fig. 7 Voltage distributions estimated from the SOC and R samples at time t = 200 s. The histograms have been normalized to match the height of the pdfs.

$$\frac{1}{C'}\int_{a_0}^{a_F}\frac{1}{a^{\gamma}}\mathrm{d}a = \int_0^{n_F} e^{\omega_s}\mathrm{d}s\,, \tag{21}$$

where a_0 and a_F are the initial and final semi-crack length, respectively, and n_F is the load cycle when the semi-crack length reaches a_F. It should be noticed that, since the random variable describing the model noise ω does not directly multiply the time (load cycle) increment dn, the integral on the right-hand side of (21) is an integral with respect to time (load cycle), and not with respect to Brownian motion. It can be approximated using the summation in Eq. (22), where the model noise is a log-Normal random variable [21]:

$$\int_0^{n_F} e^{\omega_s}\mathrm{d}s \approx \sum_{s=0}^{k-1} e^{\omega^{(i)}}\Delta n_s \quad \forall\, i = 1,\ldots,N\,, \tag{22}$$

where Δn_s is the integration step size in the load cycle domain. The log-Normal distribution of the random perturbation e^{ω} is assumed stationary, thus the subscript s can be omitted by the random variable ω in the right hand side of (22). Following the discussion in [21], we utilized a Normal distribution $\mathcal{N}(-\sigma^2/2, \sigma^2)$ to generate samples $\omega^{(i)}$. The number of elements in the summation k, similarly to (13), is the ratio between the final load cycle and the integration step size, $k = n_F\,\Delta n^{-1}$. The user can set the final crack length a_F as the desired threshold to compute the TTF pdf, or can define the time final load cycles n_F at which the crack length pdf has to be estimated. In the latter case, the solution of the Fourier's method assumes the form in (23).

$$a_F = a(n = n_F) = \left(a_0^{1-\gamma} + C'(1-\gamma)\int_0^{n_F} e^{\omega_s}\mathrm{d}s\right)^{\frac{1}{1-\gamma}} \tag{23}$$

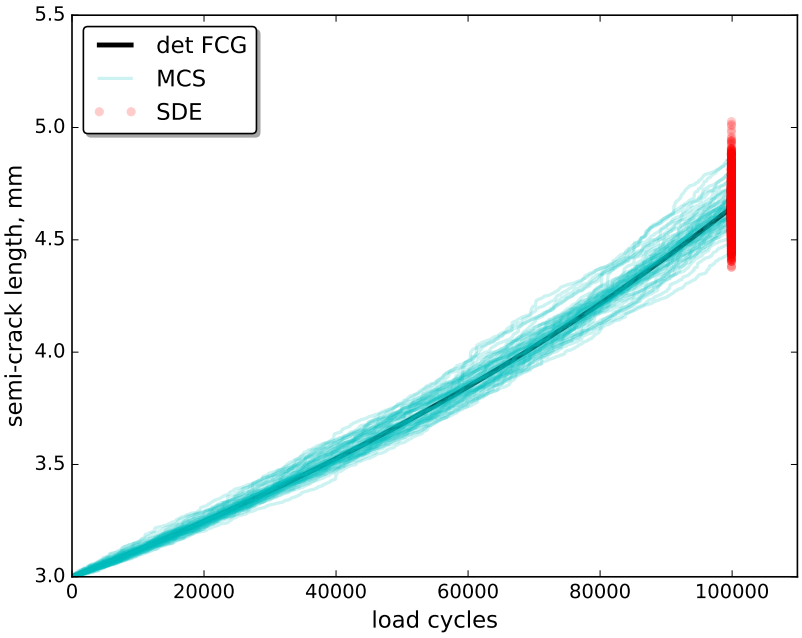

Fig. 8 Swarm of crack propagation samples propagated for n = 100,000 load cycles. Only a fraction of the MCS samples have been shown here to appreciate the different trajectories.

Table 7 Moments of the distributions calculated at n = 100,000.

$\mathbb{E}[x]$ [mm]		$\mathbb{V}[x]$ [mm^2]	
MCS	SDE	MCS	SDE
4.642	4.646	0.00878	0.00929

Figure 8 shows the simulation where the crack propagated for $n_F = 100,000$ load cycles. Only a subset of all MCS samples has been shown to make the figure readable. The exponential-like behavior of crack propagation is barely visible in this figure, given the limited size of the crack for the model parameters chosen for this case study. The pdfs at $n_F = 100000$ are presented in Fig. 9. The light blue paths are the samples generated with MCS and the red dots are the samples generated with the proposed method. Similarly to the case studies in the previous subsections, the results are analyzed in terms of similarities of distributions, moments, and t-test.

Tables 7 and 8 shows the first and second moments calculated with such distributions, as well as KL divergence and t-test statistics. The time required to draw the samples of the crack length $a^{(i)}$ using MCS and SDE method are comparable. In this case, similarly to case study 1, the SDE method does not provide any advantage in computing the desired pdf. However, as seen in the previous case study, SDE is much faster when computing the TTF pdf. The latter requires the approximation of the integral in Eq. (22), where the expected value of the TTF (in load cycles) $\mathbb{E}[n_F]$ estimated using the deterministic equation (i.e., right-hand side of Eq. (21)) substitutes n_F. The TTF pdfs are shown in Fig. 10. Similarly to the previous case studies, the pdfs calculated with the two methods are compared in Table 9.

Table 8 Similarity of the distributions at n = 100,000.

$\mathrm{KL}(p_{\mathrm{MCS}}\|\|p_{\mathrm{SDE}})$	$\tilde{t}$	$t_{\nu/2,2N-2}$	computing time [s]	
			MCS	SDE
0.00198	1.748	1.961	0.180	0.205

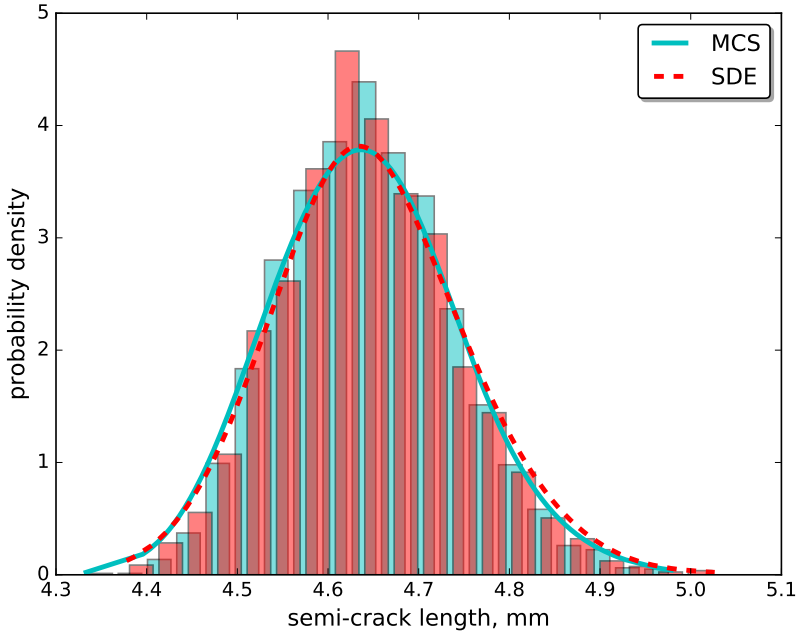

Fig. 9 semi-crack length pdf at n = 100,000 load cycles. The histograms have been normalized to match the height of the pdfs.

Table 9 Comparison between TTF distributions at a_{th} = 6 mm.

$\mathrm{KL}(p_{\mathrm{MCS}}\|\|p_{\mathrm{SDE}})$	$\tilde{t}$	$t_{\nu/2,2N-2}$	computing time [s]	
			MCS	SDE
0.00180	1.579	1.961	29.617	0.156

The proposed SDE method is hundreds of times faster than MCS when computing the TTF pdf, as already saw in the first case study. The model parameters utilized to produce the results are the following:

$$
\begin{aligned}
N &= 2500, \text{ [-]} \\
C' &= C\,(F\,\Delta L\,\sqrt{\pi})^m, \text{ [-]} \\
C &= 2.382e-12, \text{ [mm/cycle}\,(\text{MPa}\sqrt{\text{mm}})^{-m}\text{]} \\
m &= 3.2, \text{ [-]} \\
F &= 1, \text{ [-]} \\
\Delta L &= 40, \text{ [MPa] (load ratio } R = 0) \\
\gamma &= m/2, \text{ [-]} \\
\Delta n &= 100, \text{ [-]} \\
\sigma^2 &= 1.0, \text{ [-]} \\
a_0 &= 3, \text{ [mm]} \\
a_{th} &= 6, \text{ [mm]}
\end{aligned}
$$

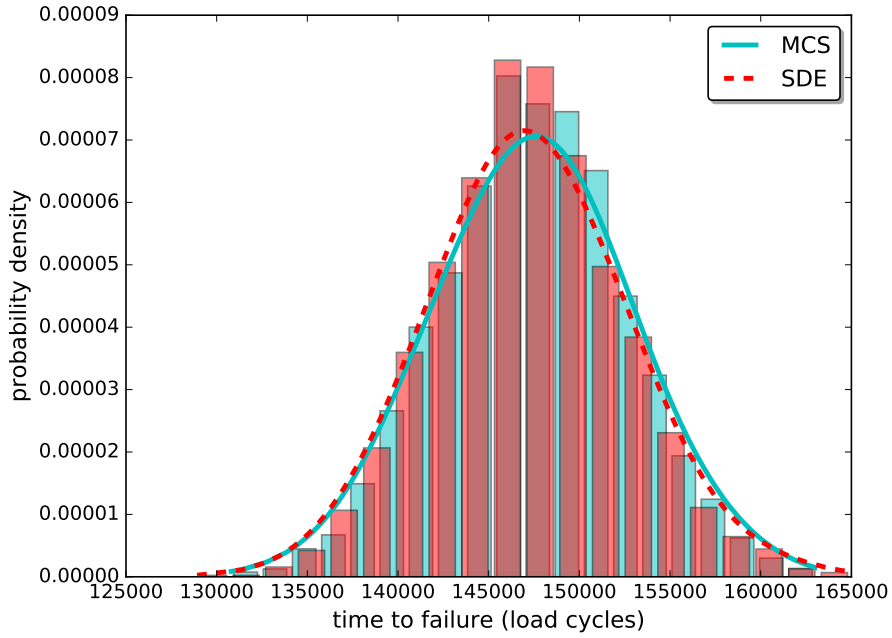

Fig. 10 Time-to-failure distribution (in load cycles) of crack propagation with limit crack length a_{th} = 6 mm. The histograms have been normalized to match the height of the pdfs.

V. Conclusions

This paper proposed a Monte Carlo solution to prognostics and long term predictions based on the properties of stochastic calculus. The similarities between the state-space formulation and stochastic differential equations are highlighted. The aim of the proposed method is to accelerate, when possible, the computation of the probability density functions of random variables described by ordinary differential equations. The work demonstrated that the proposed method, briefly defined SDE method, may save computation time when compared with full Monte Carlo simulations, briefly defined MCS, since the former does not require recursive propagation of the samples through the model equation to obtain the desired probability density functions. The distributions resulting from the two methods appear indistinguishable, being the Kullback-Leibler divergence always approaching zero, and being the mean values equal to each other according to the t-tests. A qualitative, visual comparison of the distribution has also been carried out by plotting kernel density estimations and histograms of the state variables computed with both methods. In some of the proposed case studies, the estimation of the system's state variables using MCS appeared faster than, or comparable to, the proposed SDE method. However, this never happens when computing the time-to-failure distribution, where the suggested approach is superior in terms of computing time, with no evidence of lower precision. The authors hypothesize that the simplicity of the models in case study 1 and 3 (besides the third case study focuses on a nonlinear model, the discrete formulation is a simple scalar equation) is favoring the recursive propagation of the samples against the numerical solution of the stochastic integral, which requires several random samples. This caused MCS to be faster when estimating the quantities of interest at certain time t. Further runs of the algorithms, including a series of repeated runs to statistically characterize the computing time, should be performed to quantify the computing advantage of one method over the other. In the preliminary results shown here, when SDE does not outperform MCS, the computing time of the two methods is closely comparable. On the other hand, the estimation of the time-to-failure pdf using SDE, when possible, is hundreds of time faster than the estimation performed with MCS, creating a tangible advantage to the end user who can obtain the time-to-failure distribution (and therefore, the remaining useful life of the system), in a fraction of the time required by MCS. Before the deployment of such a method to real scenarios, the solution based on stochastic calculus should be extended to cases of uncertain initial conditions, uncertain model parameters, and uncertain inputs, where x_0, $\boldsymbol{\theta}$ and $\boldsymbol{u}$ are known in terms of probability distributions. Further studies should include a sensitivity analysis of the method on the number of samples drawn during the simulations, and on the number of prediction steps. The

extension of the method to other, more complex model classes, or more complex SDEs, when possible, would enable Monte Carlo algorithms that could operate in environments with limited computing power.

References

[1] Caflisch, R. E., "Monte carlo and quasi-monte carlo methods," *Acta numerica*, Vol. 7, 1998, pp. 1–49.

[2] Goebel, K., "Prognostics, The Science of Making Prediction," *CreateSpace Independent Publishing Platform*, , No. 1, 2017.

[3] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.

[4] Zio, E., and Zoia, A., "Parameter identification in degradation modeling by reversible-jump Markov Chain Monte Carlo," *IEEE Transactions on Reliability*, Vol. 58, No. 1, 2009, pp. 123–131.

[5] Guan, X., Jha, R., and Liu, Y., "Trans-dimensional MCMC for fatigue prognosis model determination, updating, and averaging," *Annual conference of the prognostic and health management society*, 2010.

[6] He, W., Williard, N., Osterman, M., and Pecht, M., "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method," *Journal of Power Sources*, Vol. 196, No. 23, 2011, pp. 10314–10321.

[7] Orchard, M. E., and Vachtsevanos, G. J., "A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine," *Control & Automation, 2007. MED'07. Mediterranean Conference on*, IEEE, 2007, pp. 1–6.

[8] Saha, B., Goebel, K., Poll, S., and Christophersen, J., "Prognostics methods for battery health monitoring using a Bayesian framework," *IEEE Transactions on instrumentation and measurement*, Vol. 58, No. 2, 2009, pp. 291–296.

[9] Zio, E., and Peloni, G., "Particle filtering prognostic estimation of the remaining useful life of nonlinear components," *Reliability Engineering & System Safety*, Vol. 96, No. 3, 2011, pp. 403–409.

[10] An, D., Choi, J.-H., and Kim, N. H., "Prognostics 101: A tutorial for particle filter-based prognostics algorithm using matlab," *Reliability Engineering & System Safety*, Vol. 115, 2013, pp. 161–169.

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

[12] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, Vol. 50, No. 2, 2002, pp. 174–188.

[13] Higham, D. J., "An algorithmic introduction to numerical simulation of stochastic differential equations," *SIAM review*, Vol. 43, No. 3, 2001, pp. 525–546.

[14] Herzog, F., "Stochastic Differential Equations," lecture notes, 2013.

[15] Kullback, S., *Information theory and statistics*, Courier Corporation, 1997.

[16] MacKay, D. J., and Mac Kay, D. J., *Information theory, inference and learning algorithms*, Cambridge university press, 2003.

[17] Rossum, G. v., "Python tutorial, Technical Report CS-R9526," *Centrum voor Wiskunde en Informatica (CWI), Amsterdam*, 1995.

[18] Hunter, J. D., "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering*, Vol. 9, No. 3, 2007, pp. 90–95. doi:10.1109/MCSE.2007.55.

[19] Jones, E., Oliphant, T., Peterson, P., et al., "SciPy: Open source scientific tools for Python," , 2001. URL "`http://www.scipy.org/`".

[20] Celaya, J., Kulkarni, C., Biswas, G., Saha, S., and Goebel, K., "A model-based prognostics methodology for electrolytic capacitors based on electrical overstress accelerated aging," *Annual Conference of the Prognostics and Health Management Society*, 2011.

[21] Corbetta, M., Sbarufatti, C., Giglio, M., and Todd, M. D., "Optimization of nonlinear, non-Gaussian Bayesian filtering for diagnosis and prognosis of monotonic degradation processes," *Mechanical Systems and Signal Processing*, Vol. 104, 2018, pp. 305–322.

[22] Sierra, G., Orchard, M., Goebel, K., and Kulkarni, C., "Battery Health Management for Small-size Rotary-wing Electric Unmanned Aerial Vehicles: An Efficient Approach for Constrained Computing Platforms," *Reliability Engineering & System Safety*, 2018.

[23] Paris, P., and Erdogan, F., "A critical analysis of crack propagation laws," *Journal of Fluids Engineering*, Vol. 85, No. 4, 1963, pp. 528–533.

[24] Cadini, F., Zio, E., and Avram, D., "Monte Carlo-based filtering for fatigue crack growth estimation," *Probabilistic Engineering Mechanics*, Vol. 24, No. 3, 2009, pp. 367–373.

[25] Dowling, N. E., *Mechanical behavior of materials: engineering methods for deformation, fracture, and fatigue*, Prentice hall, 1993.