THE INTERACTING PARTICLE SYSTEM METHOD ADAPTED TO PIECEWISE DETERMINISTIC PROCESSES

H. CHRAIBI
A. DUTFOY
EDF R&D - Departement de management des risques industriels
7 Bd gaspard Monge, 91120 Palaiseau, France

T. GALTIER [*]
University Paris-Diderot - Laboratoire de probabilités et modèles aléatoires
8 Place Aurelie Nemours, 75013 Paris, France

J. GARNIER
Ecole Polytechnique - Centre de mathmatiques appliques
Route de Saclay, 91128 Palaiseau, France

The reliability assessment of complex power generation systems generally relies on simulation methods. When the system failure is a rare event, the MC methods are computationally intensive and we need to use a variance reduction method to accelerate the reliability assessment of such system. Among variance reduction methods, one may think of particles filters methods such as the interacting particle system method (IPS). The interest of these methods is that they do not require much knowledge about system failure to be applied, and therefore they are well suited to industrial applications. Power generation systems often follow deterministic dynamics which are altered by components' failures, components' repairs, and automatic control mechanisms. We model such dynamic hybrid systems using piecewise deterministic Markovian processes. When simulated on a short period of time, such processes tend to often generate the same deterministic trajectory, thus limiting the efficiency of the IPS method for which it is preferable to generate many different trajectories on short intervals of time. To reduce this phenomenon, we propose an adaptation of the IPS method based on the memorization method: conditioning the generated trajectories to avoid the most probable ones while computing exactly the influence of the most probable trajectories.

Keywords: Rare event - Reliability assessment - PDMP or PDP - Interacting particle system - Dynamical hybrid system - PycATSHOO

[*] Corresponding Author: tgaltier@gmail.com

1. Introduction

For both safety and regulation issues, the reliability of power generation systems has to be assessed. The considered systems (dams or nuclear plants for instance) are complex dynamic hybrid systems. Therefore, only simulation methods can be reasonably considered to assess their reliability. As the systems are highly reliable, their failure is a rare event. In this context it is well-known that the naive Monte-Carlo simulation method (MC) is time-consuming, so we want to use a variance reduction method to accelerate reliability assessment for our dynamic hybrid systems.

A failure of a dynamic hybrid system often corresponds to a physical variable of the system (temperature, pressure, water level) entering a critical region. The simulation of such a system requires an accurate model of the physical variables' trajectory. These physical variables are often determined by simple differential equations but those equations depend on the statuses of the multiple components of the systems (on, off, or failed). Thus the model should also incorporate the dynamics of the statuses of the components, which rely on deterministic feedback mechanisms and on random failures and repairs. To encounter for this hybrid interplay between the discrete process of components' statuses and the continuous evolution of the physical variables, we model the evolution of the state of a system by a piecewise deterministic Markovian process (PDMP) [2,4]. PDMPscan easily incorporate component aging, failure on demand, and delays before repairs. EDF has recently developed the PyCATSHOO tool [1], which allows the modeling of dynamic hybrid systems. PyCATSHOO evaluates the dependability criteria of the studied systems by Monte Carlo simulations. The objective of our work is to set up new algorithms to accelerate these evaluations. Here we focus on the acceleration of the reliability assessment of a system using a variance reduction methods.

We are interested in the interacting particles system method (IPS) [3], our goal being to improve its efficiency when it is applied on PDMPs. Unlike in importance sampling methods, in the IPS method we keep simulating according to the original system. The difference is that we do not simulate directly the trajectories on the entire observation duration, but we simulate the trajectories sequentially by alternating between an exploration step and a selection step. During the exploration step we simulate the trajectories on a small time interval, therefore exploring the most probable trajectories on a short horizon of time. Then we apply a selection step on these trajectories replicating the trajectories which seem close to failure and giving up the less "promising" trajectories. At the next exploration step, only replicated trajectories are continued, before the next selection, and so on... This way the effort of simulation is concentrated on trajectories which have higher chance of becoming a failing trajectory before the end of the observation duration. In the end we get more failing trajectories to fuel our estimation, and the IPS yields an unbiased estimator with a smaller variance than the MC estimator.

When we try to apply the IPS method to reliable complex hybrid systems we run into two issues: The first issue concerns the exploration: With reliable components and slow repairs there is a high probability that no component failure or repair occur during the short exploration time, and with PDMP it means that all the trajectories are likely to follow the same deterministic paths. So when we explore the trajectory space most simulated trajectories end up being the same one, hence limiting our exploration of the trajectory space. To avoid this pitfall, we propose an approach using the memorization method developed in [5]. The idea is to start the exploration by finding the most likely trajectories continuing each batch of replicated trajectories and to condition the rest of the exploration to avoid these trajectories hence simulating more different trajectories. The second issue concerns the selection step. The selection step uses a potential function that roughly measures the closeness of a trajectory to system failure. We propose a potential function adapted to multi-component systems involving control mechanisms. We have compared the efficiency of our version of the IPS method to its original version on a small system showing significant improvement in terms of variance reduction.

The rest of the paper paper is organised as follows: The section 2 is dedicated to our model of the system, the section 3 presents the IPS algorithm and of its estimator, the section 4 presents our adaptation of the IPS algorithm and of its estimator using the memorization method, and finally in section 5 we compare the efficiency of the IPS method with our adaptation based on the memorization method.

2. A model of the system based on a PDMP

Denote Z_t the state of the system at time t. Z_t is the combination of the physical variables of the system, noted X_t, and of the statuses of all the components within the system, noted M_t. Therefore $Z_t = (X_t, M_t)$. We consider that $X_t \in \mathbb{R}^d$, and that $M_t \in \mathbb{M} = \{On, Off, F\}^{N_c}$ where F corresponds to a failed status, and N_c is the number of components in the system. Within a mode $m \in \mathbb{M}$ the value of the physical variables is restricted to an open and connected set $\Omega_m \subset \mathbb{R}^d$. Defining $E_m = \{(x,m), x \in \Omega_m\}$, the set of possible states is then: $E = \bigcup\limits_{m\in\mathbb{M}} E_m$.

A discontinuity in the value of Z_t is called a jump. Between two jumps the dynamic of the system is given by an ordinary differential equation $\frac{d\,X_t}{dt} = F_{M_t}(X_t)$. We note $\phi_{(x,m)}(t)$ the solution of this equation when $X_0 = x$. Then for any time $s > 0$, if T is the time until the next jump, we have $\forall t \in [0,T)$, $Z_{t+s} = \big(X_{t+s}, M_s\big) = \big(\phi_{(X_s,M_s)}(t), M_s\big)$. Similarly, a flow function on the states can be defined. If $z = (x,m) \in E$, then $\Phi_z(t) = \big(\phi_{(x,m)}(t), m\big)$, and so $\forall t \in [0,T)$, $Z_{t+s} = \Phi_{Z_s}(t)$.

As the physical variables are often continuous, the jumps are essentially used to model changes in the statuses of the components. These jumps can occur for two reasons. Firstly, jumps can correspond to a spontaneous failure or to a repair. In such case the occurrence time of the jump is modeled using a jump rate $\lambda(Z)$. Secondly, a jump can correspond to an automatic control mechanism. We define the sets Ω_m so that the jumps associated to controls are triggered when Z_t hits the boundary of E_m. Eventually, the cdf of T (the time until the next jump starting form a state $Z_s = z$) takes this form:

$$\mathbb{P}(T \leq t | Z_s = z) = \begin{cases} 1 - \exp\left[-\Lambda_z(t)\right] & \text{if } t < t_z^*\,, \\ \qquad 1 & \text{if } t \geq t_z^*\,. \end{cases} \tag{1}$$

Here $t_z^* = \inf\{t > 0, \Phi_z(t) \in \partial E_m\}$ is the time until the flow hits the boundary starting from a state $z = (x,m)$. We take the convention that $t_z^* = +\infty$ if $\{s > 0, \Phi_z(s) \notin E_m\} = \emptyset$. If a jump occurs at time S, then the law the destination of the jump is given by a transition Kernel $K_{Z_S^-}$ where Z_S^- would be the departure state of the jump.

To generate $\mathbf{Z}_s = (Z_t)_{t\in[0,s)}$ a trajectory of the state of the system, one can repeat the following steps: Given a starting state $Z_0 = z$, generate T the time of the next jump using (1), follow the flow Φ until T, generate $Z_T = z_T$ the arrival state of the jump knowing the departure state is $Z_T^- = \Phi_z(T)$ using $K_{Z_T^-}$, and repeat starting with Z_T until you get a trajectory of size s. Defined in this way, the process Z_t is Markovian.

Remember that the goal is to estimate the probability, noted p, that the system fails before a final observation time t_f. Noting $\mathcal{A}$ the set of trajectories of length t_f which pass through the critical region we have : $p = \mathbb{P}_{z_0}\big(\mathbf{Z}_{t_f} \in \mathcal{A}\big)$.

3. The IPS algorithm and its estimator

Consider a subdivision of the interval $[0, t_f)$ into n sub-intervals of equal lengths, noted $[\tau_k, \tau_{k+1})$, and such that $0 = \tau_0 < \tau_1 < \cdots < \tau_{n\text{-}1} < \tau_n = t_f$. $\mathbf{E}_{\tau_k}$ denotes the set of trajectories defined on the k first intervals. We note Q_k the Markovian transition measure extending a trajectory $\mathbf{Z}_{\tau_k} \in \mathbf{E}_{\tau_k}$ into a trajectory $\mathbf{Z}_{\tau_{k+1}} \in \mathbf{E}_{\tau_{k+1}}$. We consider a target probability measure g_k on each sets $\mathbf{E}_{\tau_k}$ such that:

$$g_k(d\mathbf{Z}_{\tau_k}) \propto \prod_{s=1}^{k} G_s(\mathbf{Z}_{\tau_s}) \prod_{s=1}^{k} Q_s(d\mathbf{Z}_{\tau_s} | \mathbf{Z}_{\tau_{s\text{-}1}})\ \delta_{z_0}(dZ_0) \tag{2}$$

where the G_s are positive functions on $\mathbf{E}_{\tau_s}$ called the potential functions. For a measure g and a function h we note $g(h) = \int h\, dg$. For each step k the IPS provides a way too generate 2 samples of N trajectories $(\widetilde{\mathbf{Z}}^j_{\tau_k})_{j \le N}$ and $(\mathbf{Z}^j_{\tau_k})_{j \le N}$ which respectively approximate the probability measures g_k and $g_k Q_k$. These approximations are given by:

$$\widehat{g_k}(.) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\widetilde{\mathbf{Z}}^i_{\tau_k}}(.) \quad \text{and} \quad \widehat{g_{k\text{-}1} Q_k}(.) = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{Z}^i_{\tau_k}}(.) \tag{3}$$

The algorithm to build these samples is the following:

- Start with $k = 1$, and $\mathbf{Z}^j_{\tau_0} = \widetilde{\mathbf{Z}}^j_{\tau_0} = z_0$ $(\forall j = 1..N)$.
- While $k \le n$ repeat these 2 steps incrementing k each time:
  1. Simulate the trajectories $\mathbf{Z}^1_{\tau_k}, \ldots, \mathbf{Z}^N_{\tau_k}$ with
  $$\mathbf{Z}^j_{\tau_k} \sim Q_k\big(d\mathbf{Z}^j_{\tau_k} \big| \widetilde{\mathbf{Z}}^j_{\tau_{k\text{-}1}}\big)$$
  2. Create the sample of the $\widetilde{\mathbf{Z}}^j_{\tau_k}$'s by drawing with replacement in the sample of the $\mathbf{Z}^j_{\tau_k}$'s. Each trajectory $\mathbf{Z}^j_{\tau_k}$ being drawn with probability $\frac{G_k(\mathbf{Z}^j_{\tau_k})}{\sum_{i=1}^{N} G_k(\mathbf{Z}^i_{\tau_k})}$.

Noticing that $p = \mathbb{E}_{z_0}\big[\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})\big] = g_{n\text{-}1} Q_n \bigg(\frac{\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})}{\prod_{s=1}^{n\text{-}1} G_s(\mathbf{Z}_{\tau_s})} \bigg) \prod_{k=1}^{n-1} g_{k\text{-}1} Q_k\big(G_k\big)$, one can get an estimator $\hat{p}$ of p by plugging our approximations of the distributions $g_k Q_k$ in that formula, yielding :

$$\hat{p} = \widehat{g_{n\text{-}1} Q_n} \left(\frac{\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})}{\prod_{s=1}^{n-1} G_s(\mathbf{Z}_{\tau_s})} \right) \prod_{k=1}^{n-1} \widehat{g_{k\text{-}1} Q_k}\big(G_k\big). \tag{4}$$

If $\mathcal{A} \subset \operatorname{supp}\big(\prod_{s=1}^{n\text{-}1} G^*_s\big)$, this estimator is asymptotically unbiased, and satisfies a CLT : $\sqrt{N}(\hat{p} - p) \underset{N \to \infty}{\longrightarrow} \mathcal{N}(0, \sigma^2_{IPS})$, see [3].

Concerning the potential functions G_s: A valid way the select trajectories that are likely to go to the failure region would be to set $g_k(d\mathbf{Z}_{\tau_k}) \propto \mathbb{E}[\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_k}] \prod_{s=1}^{k} Q_s(d\mathbf{Z}_{\tau_s}|\mathbf{Z}_{\tau_{s\text{-}1}})\,\delta_{z_0}(dZ_0)$. This implies the potential functions have the following form : $G_s(\mathbf{Z}_{\tau_s}) = \mathbb{E}[\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_s}] \,/\, \mathbb{E}[\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_{s\text{-}1}}]$. As we do not dispose of the conditional expectations $\mathbb{E}[\mathbb{1}_{\mathcal{A}}(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_s}]$ we propose to use instead an approximation of these expectation $U(\mathbf{Z}_{\tau_s})$ based on our knowledge of the system, taking $G_s(\mathbf{Z}_{\tau_s}) = \frac{U(\mathbf{Z}_{\tau_s})}{U(\mathbf{Z}_{\tau_{s\text{-}1}})}$. For a system including similar components in parallel redundancy, we propose to set $U(\mathbf{Z}_{\tau_s}) = 1$ if $\mathbf{Z}_{\tau_s}$ has already reach the failure region once, and $U(\mathbf{Z}_{\tau_s}) = \exp\big[-\alpha\, b(Z_{\tau_s}+1)^2\big]\, L(\tau_s)$ otherwise, where L is a positive function, and $b(Z)$ indicates the number of working components within a state Z, and α is a parameter tuning the strength of the selection.

4. A modified version of the IPS using Memorization

In the IPS method, the goal of the exploration step (the step (1) in the algorithm) is to generate different trajectories in the hope of finding some trajectories with a high potential. After the k-1$^{\text{th}}$ selection step (labeled (2) in the algorithm), we get a sample of trajectories $\widetilde{\mathbf{Z}}^{j}_{\tau_{k\text{-}1}}$ which contains several replicates of the same trajectories that we note $\mathbf{z}^{(i)}_{\tau_{k\text{-}1}}$. We note i_j the index of the j^{th} trajectory in the i^{th} cluster, and $N^{(i)}_{k\text{-}1}$ the number of trajectories in this cluster. During the k^{th} exploration step, we extend $N^{(i)}_{k\text{-}1}$ times the trajectory $\mathbf{z}^{(i)}_{\tau_{k\text{-}1}}$. If we use the original kernel Q_k to do so, for reliable system we are likely to simulate many times the same trajectory. We note $\mathbf{a}^{(i)}_{\tau_k}$ this trajectory, which is the trajectory extending $z^{(i)}_{\tau_{k\text{-}1}}$ without spontaneous jump and without failures of a components. Noting H the number of jumps of $\mathbf{a}^{(i)}_{\tau_k}$ that are in the in the k^{th} sub-interval, noting s_h the time of the h^{th} jump with $s_0 = \tau_{k\text{-}1}$ and $s_{H+1} = \tau_k$ and $t_h = s_{h+1} - s_h$, and noting z_k the arrival states of those jumps we have:

$$\mathbb{P}\big(\mathbf{Z}^{i_j}_{\tau_k} = \mathbf{a}^{(i)}_{\tau_k} | \mathbf{Z}^{i_j}_{\tau_{k\text{-}1}} = z^{(i)}_{\tau_{k\text{-}1}}\big) = \prod_{h=0}^{H} \exp\Big[-\int_0^{t_h} \lambda\big(\Phi_{z_h}(u)\big)du\Big] \prod_{h=1}^{H} K_{(z^{\text{-}}_h)}(z_h).$$

For reliable systems, the jump rate λ takes low values because it is the sum of the repair rates of failed components and of the failure rates of unbroken components which are all very low. Also when a jump from $z^{\text{-}}$ to z^+ is associated to a control mechanism without failure the transition kernel $K_{z^{\text{-}}}(z^+)$ is close to one. See [1] for an example of the values of the rates and the transition kernel. For this reasons the probability of generating the trajectory $\mathbf{a}^{(i)}_{\tau_k}$ is close to 1. Therefore each of the trajectories in a cluster tends to be extend many times in $\mathbf{a}^{(i)}_{\tau_k}$, which limits the efficiency of the exploration. Assuming we have $N^{(i)}_{k\text{-}1} \geq 2$, in order to force the differentiation of the trajectories extended from the i^{th} cluster, we propose to deterministically extend the first trajectory setting $\mathbf{Z}^{i_1}_{\tau_k} = \mathbf{a}^{(i)}_{\tau_k}$, and to condition the other generated trajectories to avoid $\mathbf{a}^{(i)}_{\tau_k}$. We note V_k the Markovian kernel from $\mathbf{E}_{\tau_{k\text{-}1}}$ to $\mathbf{E}_{\tau_k}$ avoiding the most probable trajectory.

These conditioned simulations can be efficiently carried out using the memorization method, as it avoids using a rejection algorithm [5]. Consider the stopping time $\tau^{(i)}$ defined such that $\forall t < \tau^{(i)}$, $Z_t = a^{(i)}_t$ and $Z_{\tau^{(i)}} \neq \mathbf{a}^{(i)}_{\tau}$. Note that $\tau^{(i)}$ is the time at which $\mathbf{Z}_{\tau_k}$ differentiates itself from $\mathbf{a}^{(i)}_{\tau_k}$. Therefore, generate $\mathbf{Z}_{\tau_k}$ knowing it will avoid $\mathbf{a}^{(i)}_{\tau_k}$ is equivalent to generate $\mathbf{Z}_{\tau_k}$ knowing $\tau^{(i)}$ is in $[\tau_{k\text{-}1}, \tau_k)$. To simulate a trajectory $\mathbf{Z}_{\tau_k}$ avoiding $\mathbf{a}^{(i)}_{\tau_k}$, one can generate $\tau^{(i)}$ knowing $\tau^{(i)} \in [\tau_{k\text{-}1}, \tau_k)$ and set $\mathbf{Z}_{\tau^{(i)}\text{-}} = \mathbf{a}^{(i)}_{\tau^{(i)}\text{-}}$, generate $Z_{\tau^{(i)}} \neq a_{\tau^{(i)}}$, and then generate the rest of the trajectory normally until τ_k. The cdf of $\tau^{(i)} \mid \tau^{(i)} \in [\tau_{k\text{-}1}, \tau_k)$ having the following form:

$$\mathbb{P}\big(\tau^{(i)} < t \big| \tau^{(i)} \in [\tau_{k\text{-}1}, \tau_k)\big) = \frac{1 - \mathbb{P}(\mathbf{Z}_t = \mathbf{a}^{(i)}_t \big| \mathbf{Z}_{\tau_{k\text{-}1}} = \mathbf{a}^{(i)}_{k\text{-}1})}{1 - \mathbb{P}(\mathbf{Z}_{\tau_k} = \mathbf{a}^{(i)}_{\tau_k} \big| \mathbf{Z}_{\tau_{k\text{-}1}} = \mathbf{a}^{(i)}_{k\text{-}1})}$$

one can use the inverse method in order to generate realisation of $\tau^{(i)} \mid \tau^{(i)} \in [\tau_{k\text{-}1}, \tau_k)$. To access this cdf one have to memorize the trajectory $a^{(i)}_{\tau_k}$ hence the name of the method.

As we modified the extension of the trajectory, we have to re-weight accordingly the influence of the generated trajectories in the estimation to correct the induced bias, yielding the following algorithm:

- Start with $k = 1$, and $\mathbf{Z}^j_{\tau_0} = \widetilde{\mathbf{Z}}^j_{\tau_0} = z_0$, $\widetilde{W}^j_0 = \frac{1}{N}$ $(\forall j)$.
- While $k \leq n$ repeat these 2 steps incrementing k each time:
  1. For each cluster, if $N^{(i)}_{k\text{-}1} = 1$ then $\mathbf{Z}^{i_1}_{\tau_k} \sim Q_k$, else:
     Set $\mathbf{Z}^{i_1}_{\tau_k} = \mathbf{a}^{(i)}_{\tau_k}$ and $W^{i_1}_k = p_i \sum_{j=1}^{N^{(i)}_{k\text{-}1}} \widetilde{W}^{i_j}_k$,
     and $\forall j \geq 2$ simulate the trajectories $\mathbf{Z}^{i_j}_{\tau_k} \sim V_k\big(\mathbf{Z}^{i_j}_{\tau_k} | \widetilde{\mathbf{z}}^{(i)}_{\tau_{k\text{-}1}}\big)$
     and set $W^{i_j}_k = \frac{(1-p_i)}{(N^{(i)}_{k\text{-}1}-1)} \sum_{j=1}^{N^{(i)}_{k\text{-}1}} \widetilde{W}^{i_j}_k$
  2. $\forall j = 1..N$ set $\widetilde{W}^j_k = \frac{1}{N}$ and create the sample of the $\widetilde{\mathbf{Z}}^j_{\tau_k}$'s by drawing with replacement in the sample of the $\mathbf{Z}^j_{\tau_k}$'s. Each trajectory $\mathbf{Z}^j_{\tau_k}$ being drawn with probability $\frac{G_k(\mathbf{Z}^j_{\tau_k})W^j_{k\text{-}1}}{\sum_{i=1}^N G_k(\mathbf{Z}^i_{\tau_k})W^i_{k\text{-}1}}$.

Finally p is still estimated using the equation 4 but this time with $\widehat{g_{k\text{-}1}Q_k}(.) = \sum_{j=1}^{N} W_k^j \, \delta_{Z^j_{\tau_k}}(.)$. We call this modified version of the IPS the IPS+M method.

5. Results

To compare the IPS+M method with the actual IPS method and the MC method, we have applied each method on a two-components system. The system is a room heated by two heaters in passive redundancy. Heaters are programmed to maintain the temperature of the room above negative values, turning on when the temperature drops below some positive threshold and turning off when the temperature crosses a high threshold. The second heater can activate only when the first one is failed. The system fails when the temperature falls below zero. The probability p was estimated to 2.71×10^{-5} thanks to a massive Monte-Carlo of 10^7 simulations.

Table 1. Means of estimations and empirical variances on 100 runs with $N = 10^5$ for the MC, the IPS and the IPS+M methods

		MC	IPS	IPS+M
$n = 0$	$\hat{p}$	2.71×10^{-5}		
	$\hat{\sigma}^2$	2.90×10^{-10}		
$n = 5$	$\hat{p}$		2.86×10^{-5}	2.70×10^{-5}
	$\hat{\sigma}^2$		1.78×10^{-9}	1.37×10^{-10}
$n = 10$	$\hat{p}$		2.85×10^{-5}	2.64×10^{-5}
	$\hat{\sigma}^2$		1.08×10^{-9}	1.07×10^{-10}

The results of the simulation study are displayed in table 1. In this case, they highlight that the IPS method is ill-suited to PDMPs, as it yielded a higher variance than the MC method. Conversely, our IPS+M method performed well and has overcome the issue of the PDMP. Indeed, in this case, it reduced the variance by a factor 2.8 compared to the MC method, and by a factor 10 compared to the IPS method.

References

[1] H. Chraibi, J-C. Houbedine, and A. Sibler. Pycatshoo: Toward a new platform dedicated to dynamic reliability assessments of hybrid systems. PSAM congress, 2016.

[2] M.H.A. Davis. *Markov Models & Optimization*, volume 49. CRC Press, 1993.

[3] P. Del Moral. Feynman-kac formulae. In *Feynman-Kac Formulae*. Springer, 2004.

[4] F. Dufour, H. Zhang, et al. *Numerical Methods for Simulation and Optimization of Piecewise Deterministic Markov Processes*. John Wiley & Sons, 2015.

[5] P.E. Labeau. Probabilistic dynamics: estimation of generalized unreliability through efficient Monte-Carlo simulation. *Annals of Nuclear Energy*, 23(17):1355–1369, 1996.