Importance sampling for reliability assessment of Piecewise Deterministic Markov Processes

Speaker: Guillaume CHENNETIER
Joint work with: Josselin GARNIER (CMAP), Hassane CHRAIBI and Anne DUTFOY (EDF R&D)

CMAP - Ecole Polytechnique
EDF R&D

Sommaire

Piecewise Deterministic Markov Process

Introduction

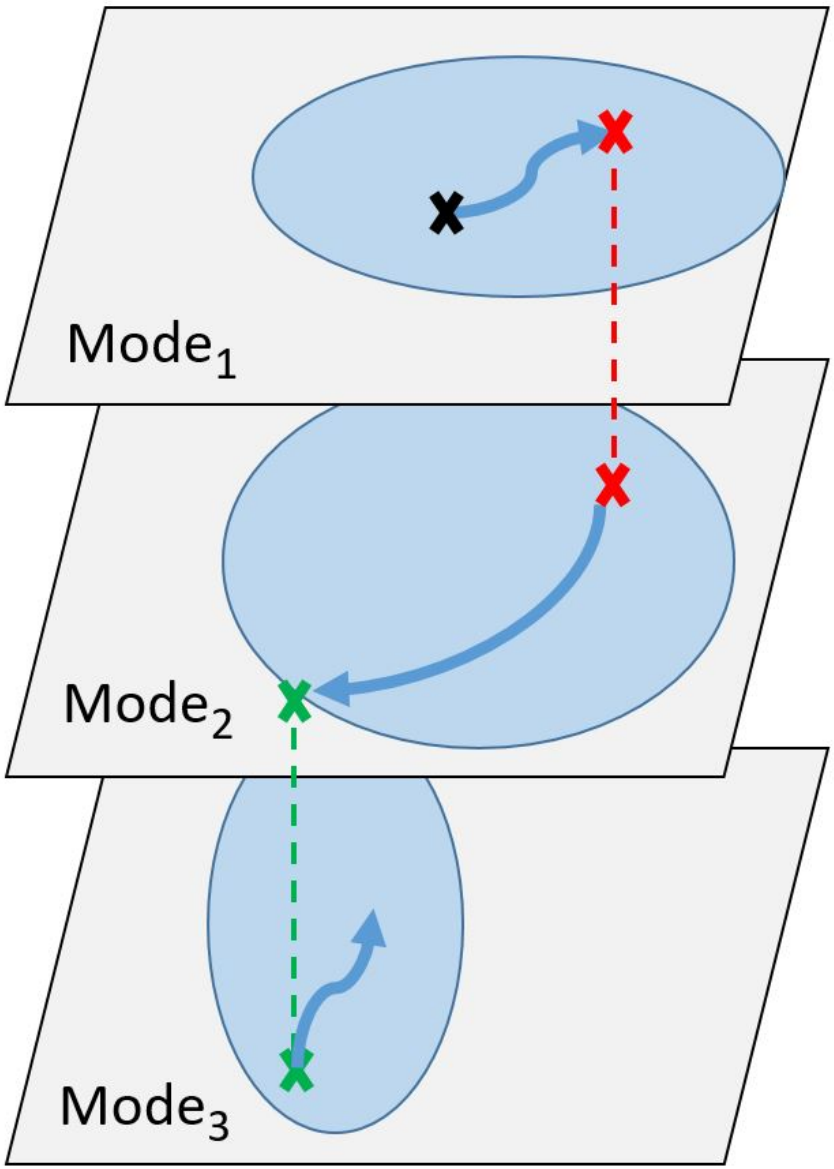
Piecewise Deterministic Markov Process
(M.H.A Davis 1984)

Hybrid process: $Z_t = (X_t, M_t)$

- position X_t is continuous
- mode M_t is discrete

Piecewise deterministic: the position follows a deterministic trajectory depending on the mode.

Markov process: at random times and deterministic times the process jumps to random states. The distribution of the next jumps does not depend on past states.



Characterization of a PDMP

The **flow function** Φ, solution of differential equations, gives the deterministic dynamic. If there is no jump between time s and time $s+t$ then:

$$Z_{s+t} = \Phi_{Z_s}(t)$$

The deterministic jumps occur when the process reaches the **boundaries of the state space** E.

$$t_z^* = \inf\{t > 0 : \Phi_z(t) \in \partial E\}$$

The **jump intensity** λ gives the distribution of the time of the next random jump.

$$\mathbb{P}(T > t \mid Z_s = z) = \mathbb{1}_{t<t_z^*} \exp\left(- \int_0^t \lambda\left(\Phi_z(u)\right) du \right)$$

The **jump kernel** $\mathcal{K}$ gives the law of the location after a jump: the process jumps from z^- to state z with probability $\mathcal{K}_{z^-}(z)$.

Likelihood

Some methods (such as importance sampling in my case) require the computation of the likelihood of a PDMP trajectory.

Recent works (Thomas Galtier 2019) explicited the dominant measure ξ (and the associated space and σ-algebra) for which a trajectory admits a probability density function.

Probability density function of a PDMP trajectory

If the PDMP $\boldsymbol{Z}$ admits n_Z jumps at time $t_1, \dots, t_{n_Z}$ in states $z_1, \dots, z_{n_Z}$, then its density f with respect to ξ is:

$$f(\boldsymbol{Z}) = \prod_{k=0}^{n_Z} \left[\lambda\left(\Phi_{z_k}(t_k)\right)\right]^{\mathbb{1}_{t_k < t^*_{z_k}}} \exp\left[-\int_0^{t_k} \lambda\left(\Phi_{z_k}(s)\right)\, ds\right] \prod_{k=1}^{n_Z} \mathcal{K}_{z_k^-}(z_k)$$

Take home message: the computation of the pdf of a PDMP trajectory is easy and does not require to recalculate the flow.

Multi-components system modeling

Purpose: estimating the probability that the physical variables (temperature, water level, pressure, etc.) of a dynamic system exceed a critical threshold. This occurs only after the degradation of certain components, which is unpredictable.

PDMP framework:

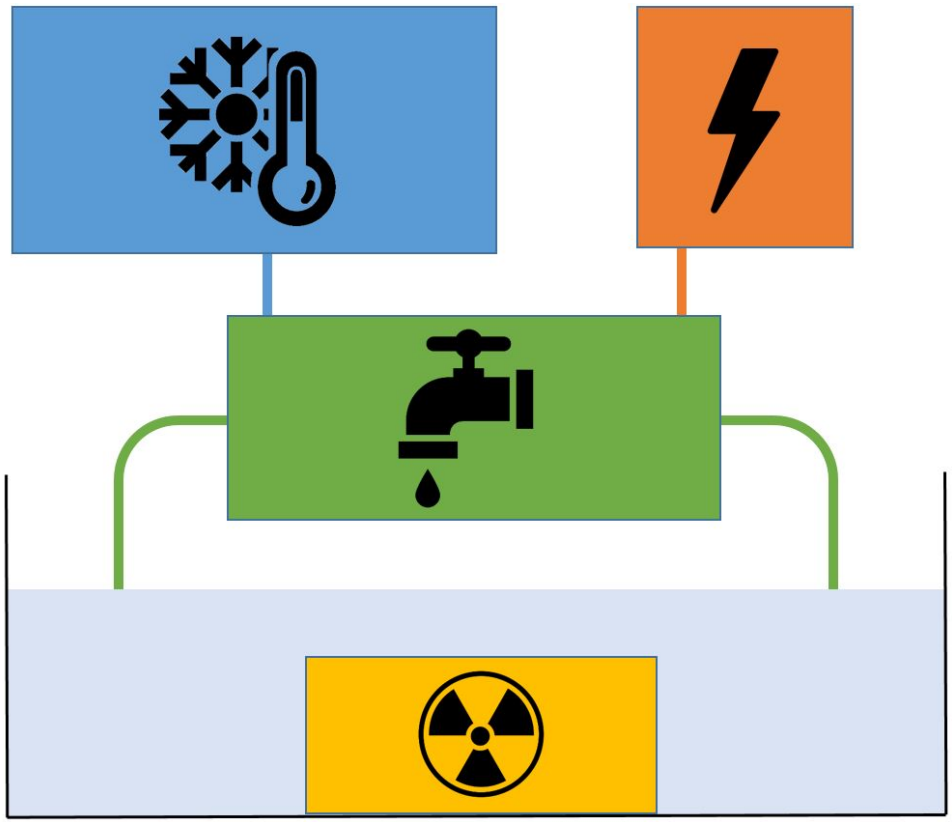
- **Deterministic flow**: induced by the physical laws governing the evolution of the variables of interest. It is computed by expensive numerical codes. This is what costs the most in a simulation.
- **Jump intensity and kernel**: built from failure rates, probability of failure on demand and repair rates of each component (provided by engineers).

`PyCATSHOO` tool by EDF performs such PDMP simulations.

Test case: the spent fuel pool

Spent nuclear fuel is stored in a pool to be cooled. To ensure the cooling of the fuel, the system must:

1. Draw cold water from outside.
2. Transfer its temperature to the pool water through sealed circuits.
3. Supply power to key components.



When the system fails to cool the fuel, the water eventually evaporates and the fuel may cause serious damage. We are looking for the probability of the water level reaching a critical threshold.

Importance sampling

The issue with crude Monte-Carlo

We would like to estimate the failure probability of the system *i.e* the probability for a trajectory of a PDMP with distribution f to reach a failure area $\mathcal{D}$.

$$P = \mathbb{P}_f(\boldsymbol{Z} \in \mathcal{D}) = \mathbb{E}_f\left[\mathbb{1}_{\boldsymbol{Z} \in \mathcal{D}}\right]$$

Crude Monte-Carlo estimator

$$P \approx \frac{1}{N} \sum_{k=1}^{N} \mathbb{1}_{\boldsymbol{Z}_k \in \mathcal{D}} = \widehat{P}_{\text{CMC}} \quad \text{with } \boldsymbol{Z}_k \sim f$$

Issue: if P is small (in our case smaller than 10^{-6}) then most of the realizations of the PDMP will fall outside of $\mathcal{D}$. A proper estimation thus requires a tremendous number of simulations.

Goal: Finding an estimator with a lower variance that can closely estimate the probability of failure with less than 10^4 simulations.

Importance sampling

Idea: generating processes from an other distribution g which produces more faulty trajectories than f then fix the bias with a likelihood ratio.

$$P = \mathbb{E}_f\left[\mathbb{1}_{z\in\mathcal{D}}\right] = \int \mathbb{1}_{z\in\mathcal{D}} f(z) d\xi(z)$$
$$= \int \mathbb{1}_{z\in\mathcal{D}} \frac{f(z)}{g(z)} g(z) d\xi(z) = \mathbb{E}_g\left[\mathbb{1}_{z\in\mathcal{D}} \frac{f(Z)}{g(Z)}\right]$$

IS estimator

$$P \approx \frac{1}{N}\sum_{k=1}^{N} \mathbb{1}_{Z_k\in\mathcal{D}} \frac{f(Z_k)}{g(Z_k)} = \widehat{P}_{\mathrm{IS}} \quad \text{with } Z_k \sim g$$

Choice of g: very delicate. Optimal choice leads to a zero variance estimator but poor choices lead to an infinite variance estimator.

Previous work: Thomas Galtier's thesis

The optimal distribution g^* of the optimal process exists:

1. it is the distribution of a PDMP...
2. with the same deterministic flow Φ than the original one!
3. it is therefore completely described by its jump intensity λ^* and jump kernel $\mathcal{K}^*$ whose expression is given by:

$$\lambda^*(\Phi_z(t), s) = \lambda(\Phi_z(t)) \times \frac{U^-\left(\Phi_z(t), s+t\right)}{U^*\left(\Phi_z(t), s+t\right)}$$

$$\mathcal{K}^*_{z^-,s}\left(z\right) = \mathcal{K}_{z^-}(z) \times \frac{U^*\left(z, s+t\right)}{U^-\left(z^-, s\right)}$$

where U^* is a commitor function:

$$U^*(z, s) = \mathbb{P}_f(\boldsymbol{Z} \in \mathcal{D} \,|\, Z_s = z)$$

$$U^-(z^-, s) = \sum_z U^*(z, s)\mathcal{K}_{z^-}(z)$$

Parametric approach and results

Importance sampling in practice

Problem: the commitor function U^* is not known explicitly.

We replace it by an approximation $U_{\boldsymbol{\alpha}}$ where $\boldsymbol{\alpha}$ denotes a vector of parameters that can be adjusted during estimation procedure to improve the approximation. $U_{\boldsymbol{\alpha}}$ should quantify the distance between any state z and the failure area $\mathcal{D}$.

Importance process: trajectories are simulated under distribution $g_{\boldsymbol{\alpha}}$ with jump intensity $\lambda^{\boldsymbol{\alpha}}$ and jump kernel $K^{\boldsymbol{\alpha}}$ obtained by replacing U^* by $U_{\boldsymbol{\alpha}}$ in the expressions of U^-, λ^* and K^*.

Optimization by Cross-Entropy: once the parametric family $(U_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}}$ is chosen, it remains to find the best value for parameter $\boldsymbol{\alpha}$. It is set iteratively by the Cross-Entropy method.

Cross Entropy

Idea: we look for the closest density to g^* into a parametric family $(g_\alpha)_{\alpha \in A}$ according to the Kullback Leibler divergence. The resulting minimization program depends on f, $\mathcal{D}$ and $g_{\boldsymbol{\alpha}}$ but not on g^*!

$$\begin{aligned} \arg\min_{\boldsymbol{\alpha} \in A} \mathrm{KL}\left(g^* \| g_{\boldsymbol{\alpha}}\right) &= \arg\min_{\boldsymbol{\alpha} \in A} \mathbb{E}_{g^*}\left[\log\left(\frac{g^*(\boldsymbol{Z})}{g_{\boldsymbol{\alpha}}(\boldsymbol{Z})}\right)\right] \\ &= \arg\min_{\boldsymbol{\alpha} \in A} \left\{-\mathbb{E}_f\left[\mathbb{1}_{\boldsymbol{Z} \in \mathcal{D}} \log\left(g_{\boldsymbol{\alpha}}\left(\boldsymbol{Z}\right)\right)\right]\right\} \end{aligned}$$

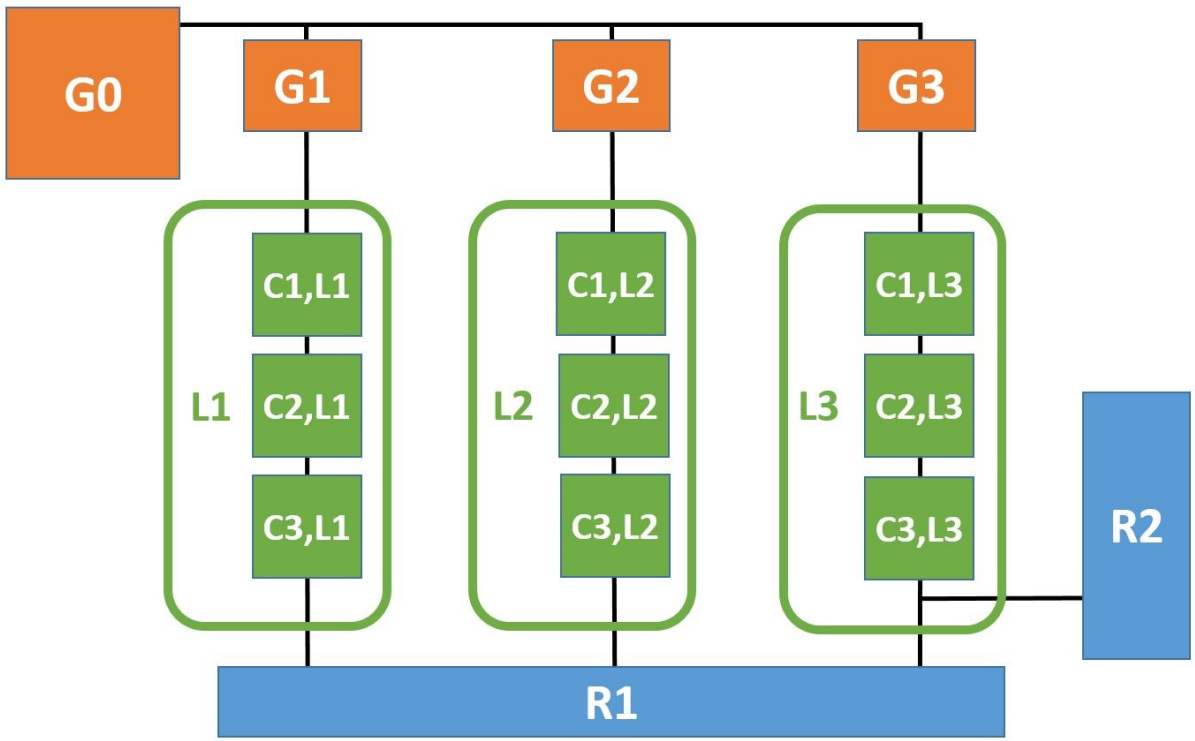
Sequential optimization: Start with an initial $\boldsymbol{\alpha}_0$ then at iteration t

1. Generate $\boldsymbol{Z}_1^{(t)}, \ldots, \boldsymbol{Z}_N^{(t)} \sim g_{\boldsymbol{\alpha}^{(t)}}$
2. Compute the solution to the below optimization problem

$$\boldsymbol{\alpha}^{(t+1)} = \arg\min_{\boldsymbol{\alpha} \in A} \left\{ -\frac{1}{N} \sum_{k=1}^{N} \mathbb{1}_{\boldsymbol{Z}_k^{(t)} \in \mathcal{D}} \frac{f\left(\boldsymbol{Z}_k^{(t)}\right)}{g_{\boldsymbol{\alpha}^{(t)}}\left(\boldsymbol{Z}_k^{(t)}\right)} \log\left(g_{\boldsymbol{\alpha}}\left(\boldsymbol{Z}_k^{(t)}\right)\right) \right\}$$

Minimal groups: the spent fuel pool case

Minimal groups: smallest sets of components that if left broken ensure system failure. (permanent repair of one component in each group prevents the failure)



Examples: $(G_0, G_1, G_2, G_3), (R_1, R_2), (C_1L_1, C_3L_2, C_1L_3)$

In this system: there is 69 minimal groups for 15 components.

Parametric family

Let d be the number of minimal groups of the system. For any state z we denote $\beta_i(z)$ the proportion of broken components in the i-th most damaged minimal group, then for $\boldsymbol{\alpha} \in \mathbb{R}_+^{d+1}$ we set:

$$U_{\boldsymbol{\alpha}}(z) = \left(\sum_{i=1}^{d} U_{\alpha_i}(z) \right)^{\alpha_0} \quad \text{with} \quad U_{\alpha_i}(z) = \alpha_i^{\beta_i(z)}$$

The U_{α_i} functions are convex and increasing in the number of broken components within their group.

- α_0 small: trying to get a failing minimal group quickly.
- α_0 large: trying to increase multiple failures in as many minimal groups as possible.

Dimension reduction: we are able to find a good $\boldsymbol{\alpha}$ value in a lower dimension space than $d+1$ during the Cross-Entropy. (outside the scope of this talk)

Experiment and results

We tested this importance sampling approach on the spent fuel pool case and we compared it to a massive crude Monte-Carlo method.

Method	N	$\widehat{P}$	95% Confidence interval
CMC	10^6	2×10^{-6}	$[-7.72 \times 10^{-7} ; 4.78 \times 10^{-6}]$
CMC	10^7	3.9×10^{-6}	$[2.67 \times 10^{-6} ; 5.12 \times 10^{-6}]$
CMC	10^8	3.50×10^{-6}	$[3.13 \times 10^{-6} ; 3.87 \times 10^{-6}]$
IS	10^2	3.91×10^{-6}	$[8.08 \times 10^{-7} ; 7.01 \times 10^{-6}]$
IS	10^3	3.03×10^{-6}	$[1.09 \times 10^{-6} ; 4.95 \times 10^{-6}]$
IS	10^4	3.35×10^{-6}	$[2.65 \times 10^{-6} ; 4.04 \times 10^{-6}]$

Table 1: Comparison between method CMC and IS

Our IS method performs as well with 10^3 to 10^4 simulations than crude Monte-Carlo method with 10^7 to 10^8 simulations. It is thus more than a thousand times more efficient.

Conclusion

Future work:

- Finding a good way to initialize the Cross-entropy method.
- Testing the method on larger systems with more subtle/complex dynamics.
- Analyzing the sensitivity of the probability of failure to the PDMP parameters without simulating new trajectories.
- Using our approximation of the commitor function U^* to estimate the probabilities of different rare sub-events with an interacting particle system method.

Thank you for your attention!