



HAL
open science

Accelerated Monte-Carlo methods for Piecewise Deterministic Markov Processes for a faster reliability assessment of power generation systems within the PyCATSHOO toolbox

Thomas Galtier

► To cite this version:

Thomas Galtier. Accelerated Monte-Carlo methods for Piecewise Deterministic Markov Processes for a faster reliability assessment of power generation systems within the PyCATSHOO toolbox. Computation [stat.CO]. Université de Paris, 2019. English. NNT : . tel-02327135

HAL Id: tel-02327135

<https://theses.hal.science/tel-02327135v1>

Submitted on 22 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Paris
Ecole doctorale de Sciences Mathématiques
de Paris Centre (ED 386)
LPSM

**Accelerated Monte-Carlo methods for
Piecewise Deterministic Markov Processes**

*for a faster reliability assessment of power generation
systems within the PyCATSHOO toolbox*

Par Thomas GALTIER

Thèse de doctorat en Statistique

Dirigée par Josselin Garnier

Présentée et soutenue publiquement le 20/09/2019

Devant le jury suivant :

Rapportrice : Benoite De Saporta, professeure, université de Montpellier
Rapporteur : Marko Cepin, professeur, université de Ljubljana
Examineur : Randal Douc, professeur, Telecom sudparis
Examineur : Stephane Boucheron, professeur, université paris diderot
Examineur : Nicolae Brinzei, maitre de conference, université de Lorraine
Encadrant : Hassane Chraibi, ingénieur, EDF
Encadrante : Anne Dutfoy, ingénieure, EDF



Except where otherwise noted, this is work licensed under <https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>

Résumé :

Cette thèse porte sur l'estimation de la fiabilité des centrales hydrauliques et nucléaires construites et exploitées par EDF (Électricité de France). La défaillance d'une centrale étant associée à des conséquences majeures (crue, rupture de barrage, ou fusion du cœur), pour des raisons réglementaires et de sûreté EDF doit s'assurer que la probabilité de défaillance d'une centrale est suffisamment faible.

La défaillance de tels système intervient lorsque des variables physiques (température, pression, niveau d'eau) dépassent un certain seuil critique. Typiquement, ces variables entrent dans cette région critique seulement lorsque plusieurs composants du système sont détériorés. Donc pour estimer la probabilité de défaillance du système, nous devons modéliser conjointement le comportement des composants et celui des variables physiques. Pour ce faire nous utilisons un modèle basé sur un Processus Markovien Déterministe par Morceaux (PDMP).

Ce modèle permet d'estimer la probabilité de défaillance du système par simulation. Malheureusement le modèle demande d'importantes ressources de calcul pour être simulé. La méthode classique d'estimation par Monte-Carlo, qui demande beaucoup de simulations pour estimer les probabilités d'événements rares, est alors beaucoup trop lente à exécuter dans notre contexte. Il est nécessaire d'utiliser des méthodes faisant appel à moins de de simulations pour estimer la probabilité de défaillance du système : des méthodes de réduction de variance.

Parmi les méthodes de réduction de variance on distingue les méthodes de type "tirage préférentiel" (importance sampling) et les méthodes de type "splitting", mais ces méthodes présentent des difficultés lorsqu'elles sont employées avec des PDMPs.

En effets les fondements théoriques du tirage préférentiel avec les PDMPs sont encore à définir. Pour être utilisé le tirage préférentiel demande de pondérer les simulations par un rapport de vraisemblance, et, pour les trajectoires de PDMPs, ce rapport de vraisemblance n'a pas pas encore été rigoureusement défini jusqu'ici. Aussi des processus d'importance efficaces pour les PDMP sont encore à définir. Cette thèse propose une façon de construire rigoureusement le rapport de vraisemblance, puis étudie les caractéristique du processus d'importance optimal pour les PDMPs, elle en déduit une façon pratique de spécifier un processus d'importance efficace. Cette méthode est testée sur des systèmes de différentes tailles, les résultats montrent de très bonnes performances sur

de petits systèmes, mais sur des systèmes de tailles industrielles la méthode proposée reste encore difficile à mettre en œuvre.

Mots clefs : tirage préférentiel – PDMP – filtre particulière – fiabilité

Abstract :

This thesis deals with the reliability assessment of nuclear or hydraulic power plants, which are built and exploited by the company EDF (Électricité de France). As the failures of such systems are associated to major human and environmental consequences, for both safety and regulatory reasons, EDF must ensure that the probability of failure of its power generation systems is low enough.

The failure of a system occurs when the physical variables characterizing the system (temperature, pressure, water level) enter a critical region. Typically, these physical variables can enter a critical region only when a sufficient number of the basic components within the system are damaged. So, in order to assess the probability of having a system failure, we have to jointly model the evolution of the physical variables and of the statuses of the components. To do so we use a model based on piecewise deterministic Markovian processes (PDMPs).

This model allows to estimate the probability of failure of the system by simulation. Unfortunately the model is computationally intensive to run, and the classic Monte-carlo method, which needs a lot of simulations to estimate the probability of a rare event, is then too computationally intensive in our context. Methods requiring less simulations are needed, like for instance variance reduction methods.

Among variance reduction methods, we distinguish importance sampling methods and splitting methods. The difficulty is that we need to use these methods on PDMPs, which raises a few issues.

The theoretical foundations for the importance sampling methods with PDMPs are yet to be defined. Indeed these methods require to weight the simulations with likelihood ratios, and these likelihood ratios have not been properly defined so far for PDMP trajectories, which are degenerate processes. Also efficient biasing strategies (i.e. altered simulation processes yielding a small variance estimator) have not been proposed for PDMPs. This thesis presents how to build the likelihood ratios, it investigates the characteristics of the ideal optimal biasing strategy, and it presents a convenient and efficient way to specify practical biasing strategies for systems of reasonable size.

Concerning particular filters methods, they tend to perform poorly on PDMPs with low jump rates and therefore they need to be adapted in order to be successfully applied to reliable power generation systems. Indeed in this context, splitting methods are sometimes less efficient than the naive Monte-Carlo method. This thesis investigates how it is possible to efficiently use these methods with PDMPs. Namely we propose an adaptation of the interacting particles system method (IPS) for PDMPs with low jump rates, and we investigate the convergence properties of the estimators of our methods. The efficiency of the method is tested on a reasonable size system showing a performance slightly better than or equivalent to the Monte-Carlo method.

An additional result on the IPS method is also proposed in a general Markovian framework (beyond PDMPs). The IPS method takes as input certain potential functions that directly impact the variance of the estimator. In this PhD, we show that there are optimal potential functions for which the variance is minimized and we give their closed-form expressions.

Keywords : importance sampling- PDMP – interacting particle system – reliability assessment

Remerciements

Ces quatre années de thèses m'ont beaucoup appris. J'en ressors grandi tant sur le plan professionnel que personnel. Pour cela je tiens à remercier les personnes qui m'ont aidé dans l'accomplissement de cette thèse.

Mes premiers remerciements vont à mon directeur de thèse Josselin Garnier, pour ses aiguillages avisés, ses relectures patientes avec mon orthographe, et sa compréhension dans les moments difficiles. La rigueur mathématique et la bienveillance de Josselin sont un exemple que je suivrai à l'avenir.

Je remercie chaleureusement mes encadrants d'EDF : Hassane Chraïbi et Anne Dutoy. Hassane, merci pour ta gentillesse exceptionnelle, et ton soutien dans les moments difficiles. Cela a beaucoup compté pour moi.

Je tiens à remercier aussi tout ceux qui ont simplement été des amis durant cette thèse : tout les thésards ou collègues du LPSM ou de PERICLES avec qui j'ai pu partager des moments agréables durant les longues pauses café ou les déjeuners. Un gros Merci à ma fanfare les Zèbra Rayures qui m'ont apporté tant de rire et de bonheur durant ces années.

Enfin je remercie ma famille pour son soutien sans faille, et particulièrement merci à mon frère Mathieu, qui m'a souvent bien rassuré et encouragé.

Contents

I	Introduction	1
1	Piecewise Deterministic Markov Processes	3
1.1	Markovian objects	3
1.1.1	Markov chains	3
1.1.2	Markov processes	6
1.1.3	Poisson Processes	8
1.1.4	Compound Poisson process	11
1.1.5	Continuous time Markov chains	12
1.2	Piecewise Deterministic Markov Processes	14
1.2.1	Generalities on first order ordinary differential equations	14
1.2.2	PDMPs without boundary	15
1.2.3	PDMPs with boundaries	21
1.2.4	Markovian object as PDMPs	26
1.3	Model a multi-component system with a PDMP	29
1.3.1	Model the evolution of the state of the system	29
1.3.2	A model for reliability assessment	30
1.3.3	The model	31
1.3.4	Example of the Heated room system	36
1.3.5	Example of the spent fuel pool	38
2	Monte-Carlo methods for rare events	43
2.1	The Monte-Carlo method and its rare event issue	43
2.1.1	Computational burden of the Monte-Carlo estimator	45
2.1.2	The principle of variance reduction method	45
2.2	Importance Sampling	46
2.2.1	Principle	46
2.2.2	The Importance Sampling estimator	47
2.2.3	Dynamical importance sampling	48
2.2.4	Variance and optimal density	49
2.2.5	Important practical concerns	51
2.3	Variance optimization methods for importance sampling	52
2.3.1	Variance optimization: the Cross-Entropy method	53
2.3.2	Important remark on the Cross-Entropy	54
2.3.3	Adaptive Cross-Entropy	55

2.4	The interaction particle method	57
2.4.1	A Feynman-Kac model	58
2.4.2	The IPS algorithm and its estimator	59
2.4.3	Estimate the variance of the IPS estimator	62
2.4.4	Classical improvements of the IPS method	64
II	Importance Sampling with PDMPs	67
3	Theoretical foundation for Importance sampling on PDMP	71
3.1	Prerequisite for importance sampling	71
3.1.1	The law of the trajectories	72
3.1.2	The dominant measure and the density	74
4	The practical and optimal importance processes	77
4.1	Admissible importance processes	77
4.2	A way to build an optimal importance process	79
4.3	Remarks on the optimal process	81
4.4	Practical importance processes for reliability assessment	83
4.5	A parametric family of importance processes for reliability assessment	84
5	Applications to power generation systems	87
5.1	Application to the Heated-Room system	87
5.1.1	A parametric family of importance processes for the Heated-Room system	87
5.1.2	Results	89
5.2	Practical issues with power generation systems	92
5.2.1	Specify a parametric family of proximity functions	92
5.2.2	Results with the series/parallel parametric proximity function on the spent-fuel-pool system	99
6	Conclusion on the importance sampling for PDMP	101
III	A contribution to the IPS method	105
7	The optimal potential functions	109
7.1	The potentials used in the literature	110
7.2	The optimal potential	111
7.3	A short simulation study for the comparison of potentials	114
7.3.1	First example	114
7.3.2	Second example	116

8	Conclusions and implications	119
IV	The interacting particle method for PDMPs	121
9	The inefficiency of IPS on concentrated PDMP	123
9.1	IPS with PDMP	124
9.1.1	A Feynman-Kac model	126
9.1.2	The IPS algorithm and its estimators	128
9.1.3	Variance estimation for the PDMP case	129
9.1.4	The SMC improvement for PDMP	131
9.2	Concentrated PDMP make the IPS inefficient	132
9.2.1	The kind of PDMP used in the reliability analysis	132
9.2.2	Concentrated PDMP	133
9.2.3	A poor empirical approximation of the propagated distributions within the IPS	134
10	Efficient generation of the trajectories using the Memorization method	137
10.1	Advantage of Memorization over a rejection algorithm	137
10.2	Principle of the memorization method	137
11	The IPS+M method for concentrated PDMPs	141
11.1	Modify the propagation of clusters	141
11.2	Convergence properties of the IPS+M estimators	144
12	Application to two test systems	151
12.1	Empirical confirmation	151
12.1.1	The Heated-room system	151
12.1.2	Results of the simulations on the Heated-room system	152
12.1.3	Remark on the SMC with Memorization	153
12.1.4	A dam system	153
12.1.5	Results of the simulations for the dam system	153
13	Conclusion on the IPS+M	155
V	Conclusion and prospects for future work	157
	Optimal intensity's expression, and some properties of U^*	163
A.1	Proof of Equality (4.11)	163
A.2	Proof of theorem 18	164
A.3	Equality (4.12)	164

Bibliographie

167

Context and positioning

This thesis deals with the *reliability assessment* of nuclear, or hydraulic, power plants, which are built and exploited by the company EDF (Électricité de France). As the failure of such systems is associated to major human and environmental consequences, for both safety and regulatory reasons, EDF must ensure that the probability of failure of its power generation systems is low enough.

The failure of a system occurs when the physical variables characterizing the system enter a critical region. For instance, a nuclear power plant fails when the temperature of its core exceeds a fusion threshold, a dam fails when the water level exceeds the overflowing threshold, or when the pressure on evacuation valves is so high that they open spontaneously.

Typically, these physical variables can enter a critical region only when a sufficient number of the basic components within the system are damaged. So, in order to assess the probability of having a system failure, we have to jointly model the evolution of the physical variables and of the statuses of the components. We define the state of the system as the pair of the physical variables and of the statuses of the components. At a time t , the state of the system is denoted by

$$Z_t = (X_t, M_t),$$

where X_t denotes the physical variables at time t , and M_t denotes the statuses of all the components in the system at time t . We consider that each of the N_c components within the system can be in a status ON, OFF, or "F" for failed, so M_t takes discrete values in $\{\text{ON}, \text{OFF}, \text{F}\}^{N_c}$, whereas the physical variables are continuous and evolve in \mathbb{R}^d . Note the state of the system Z_t is made of both continuous and discrete coordinates, such a variable is sometimes referred to as an "*hybrid variable*" in the reliability community.

The evolution of the state of the system Z_t can be modeled by a *piecewise deterministic Markovian process* (PDMP). Indeed, these processes are well suited to model the evolution of hybrid variables such as Z_t . When they were introduced, PDMPs were meant to gather all the Markovian processes that do not include diffusion, as such they benefit from very high modeling capacities. They can model the evolution of the state of most of the power generation systems.

Once we have modeled a given power generation system, we want to assess its probability of failure. More precisely we want to assess the probability, denoted p_D , that the state of the system Z_t enters the failure region D before a given observation time t_f and

after being initiated in a safe state z_0 . We are interested in computing p_D in order to assess the reliability of the system, denoted r , and defined by:

$$r = 1 - p_D. \quad (1)$$

Denoting $\mathbf{Z} = (Z_s)_{s \in [0, t_f]}$ a trajectory of the state of the systems of size t_f , and $\tau_D = \inf\{s > 0, Z_s \in D\}$ the time at which the state of systems enters D , p_D is defined by

$$p_D = \mathbb{P}(\tau_D < t_f | Z_0 = z_0). \quad (2)$$

Or denoting by \mathcal{D} the set of trajectories hitting the region D before t_f , the probability of failure p_d can be defined by:

$$p_D = \mathbb{P}(\mathbf{Z} \in \mathcal{D} | Z_0 = z_0). \quad (3)$$

Due to the complexity of the considered system (and of the associated PDMP model), it is not possible to compute this probability analytically, so we have to rely on simulation techniques. The company EDF has recently developed the PyCATSHOO toolbox¹, which allows simulating and modeling such power generation systems. Thanks to the well-known Monte Carlo method, it evaluates dependability criteria, among which is the reliability of the system.

Unfortunately, when the Monte-Carlo method is used to assess the probability of a rare event like system failure, it is computationally intensive. Indeed, if we want to identify the order of magnitude of p_D , the Monte-Carlo method requires a huge number of simulations, and it becomes excessively time consuming. Typically the assessment of a probability $p \simeq 10^{-n}$ with the Monte-Carlo method requires $N = 10^{n+2}$ simulations, which takes too much time to run. In order to reduce computation times, we need an estimator that is more accurate and requires less simulations. Typically, in our industrial setting, the question is somehow to ensure with few thousand samples that a system is unsafe with a probability p_D lower than 10^{-6} .

At this stage, a variance reduction method can provide a more accurate estimator. In such a variance reduction method, we generally increase the probability of simulating system failure by altering the simulation process, and then we correct the induced bias by an appropriate weighting of each simulation. Among variance reduction methods, one can think of two main groups of methods: the importance sampling methods and the particle filtering methods. The difficulty is that we need to use these methods on PDMPs, which raises a few issues.

1. See the website <http://pycatshoo.org/>

The theoretical foundations for the importance sampling methods with PDMP are yet to be defined. Indeed these methods require to weight the simulations with likelihood ratios, and these likelihood ratios have not been properly defined so far for PDMP trajectories, which are degenerate processes. Also efficient biasing strategies (i.e. altered simulation processes yielding a small variance estimator) have not been proposed for PDMP. This thesis presents how to build the likelihood ratios, and it presents a convenient and efficient way to specify biasing strategies for systems of reasonable size.

Concerning particular filters methods, they tend to perform poorly on PDMP with low jump rates and therefore they need to be adapted in order to be successfully applied to power generation systems. This thesis investigates how it is possible to efficiently use this method with PDMP. Namely we propose an adaptation of the importance sampling method and of a particular filter method for PDMPs, and we investigate the convergence properties of the estimators of our methods.

List of Symbols

\bar{A}	Closure of the set A
$\overset{\circ}{A}$	Interior of the set A
∂A	Boundary of the set A
$\mathcal{M}(A)$	Set of the measurable functions defined on the set A
$\mathcal{B}(A)$	Set of the Borelians of the metric space A
E	Set of the possible states of the PDMP $(Z_t)_{t \geq 0}$
E_m	Set of the possible states with mode equal to m
Ω_m	Set of the possible positions when the mode is equal to m
\mathcal{E}	σ -algebra of E
t_z^*	Hitting time of the boundary ∂E_m with the flow Φ_z
\mathcal{F}_v	The σ -algebra generated by $(Z_t)_{t \leq v}$
\mathcal{F}_{v^-}	The σ -algebra generated by $(Z_t)_{t < v}$
$K_z : E \rightarrow \mathbb{R}^+$	density of \mathcal{K}_z
$\mathcal{K}_z : \mathcal{E} \rightarrow [0, 1]$	Jump kernel when the departure state is $z \in \bar{E}$
M_t	Mode of a PDMP at time t
\mathbb{M}	Set of the possible modes of the PDMP $(Z_t)_{t \geq 0}$
S_k	Time of the k -th jump
T_k	Time between the k -th jump and the $k + 1$ -th jump
X_t	Position of a PDMP at time t
d	Dimension of the position
$\lambda : E \rightarrow \mathbb{R}^+$	State-related jump rate function
$\lambda_z : [0, t_z^*) \rightarrow \mathbb{R}^+$	Time-related jump rate function $\lambda_z = \lambda \circ \Phi_z$
$\mathbf{Z}_t = (Z_s)_{s \in [0, t]}$	Trajectory of the Z_s on the interval $[0, t]$
$\phi_m(x, \cdot) : \mathbb{R} \rightarrow \Omega_m$	Flow function on the positions
Φ_z	Flow function on the states

CONTENTS

$n(\mathbf{Z}_t)$ = Number of jumps in the trajectory \mathbf{Z}_t

$Z_t = (X_t, M_t)$ State of a PDMP at time t

PART I

Introduction

Piecewise Deterministic Markov Processes

The motivation of this thesis is to provide efficient simulation methods for the assessment of the reliability of power generation systems. The prerequisite to any simulation method, is to have a probabilistic model for the simulation output. This chapter aims at presenting how we can model and simulate the evolution of the state of a power generation system by using a Piecewise Deterministic Markovian Process (PDMP).

PDMPs benefit from high modeling capacities and can model many industrial systems. When they were first introduced by M.H.A Davis [19], this class of processes were meant to include all the Markovian objects that do not include diffusion. Therefore most of the classical Markovian objects are included in the class of PDMP. This is noteworthy because results derived for PDMPs stay true for any of these processes within the class of PDMP. For instance the results of this thesis, that are presented in part II, are adapted for PDMP and therefore can be generalized to Markov chains, to continuous time Markov chains, to compound Poisson processes, and to queuing models and to any object that can be viewed as a PDMP. To begin this Chapter we quickly present some generalities on Markov chains and Markov processes, and then we present the classical Markovian objects that are included in the PDMP class. We present the different types of PDMP models in the second section of this chapter. A third section presents how we can model a power generation system with this model. Most of the notions presented in these sections come from the books [18] and [54].

1.1 Markovian objects

The notations introduced here are used only in this section and in the subsection 1.2.4 which present how to specify some Markovian objects such as PDMP.

1.1.1 Markov chains

Definition 1 (Markov chain, Homogeneity). Let $(A, \mathcal{A}, \mathbb{P})$ be a probability space and let $(\mathbb{M}, \mathcal{M})$ be a measurable space. The sequence $(M_n)_{n \in \mathbb{N}}$ of random variables from $(A, \mathcal{A}, \mathbb{P})$

to $(\mathbb{M}, \mathcal{M})$ is a *Markov chain* if it verifies the following memory-less property:

$$\begin{aligned} &\forall B \in \mathcal{M}, k \in \mathbb{N}, n \in \mathbb{N}^*, \\ &\mathbb{P}(M_{n+k} \in B \mid M_{n-1}, M_{n-2}, \dots, M_0) = \mathbb{P}(M_{n+k} \in B \mid M_{n-1}). \end{aligned} \quad (1.1)$$

Furthermore the chain is said to be homogeneous if the probability $\mathbb{P}(M_{n+k} \in B \mid M_{n-1})$ does not depend on the time n .

Definition 2 (Filtration). The σ -algebra generated by the random variables $(M_k)_{k \leq n}$ is denoted by \mathcal{F}_n^M . These σ -algebras form the natural *filtration* of the Markov chain $(M_k)_{k \in \mathbb{N}}$.

The memory-less property displayed in equation (1.1) can then be rewritten using the filtration:

$$\begin{aligned} &\forall B \in \mathcal{M}, k \in \mathbb{N}, n \in \mathbb{N}^*, \\ &\mathbb{P}(M_{n+k} \in B \mid \mathcal{F}_{n-1}^M) = \mathbb{P}(M_{n+k} \in B \mid M_{n-1}). \end{aligned} \quad (1.2)$$

An important notion for Markovian objects is the stopping time. Such stopping times are convenient to work with, because they allow to use the corollary of strong Markov property presented below .

Definition 3 (Stopping time). A random variable T with value in $(\mathbb{N} \cup \{\infty\}, \mathcal{P}(\mathbb{N} \cup \{\infty\}))$ is a *stopping time* for the Markov chain $(M_k)_{k \in \mathbb{N}}$, if, for all $n \in \mathbb{N}$, the event $\{T \leq n\}$ belongs to the filtration \mathcal{F}_n^M .

Definition 4 (σ -algebra on a stopping time). If T is a stopping time, one can define a σ -algebra on this stopping time by

$$\mathcal{F}_T^M = \left\{ B \in \mathcal{F}_\infty^M = \sigma \left(\bigcup_{n \in \mathbb{N}} \mathcal{F}_n^M \right) \mid B \cap \{T \leq n\} \in \mathcal{F}_n^M, \forall n \geq 0 \right\}. \quad (1.3)$$

Theorem 1 (Corollary of the Strong Markov property). *A Markov Chain $(M_k)_{k \in \mathbb{N}}$ verifies the strong Markov property. As a consequence for a finite stopping time T (such that $\mathbb{P}(T < \infty) = 1$) we have*

$$\forall B \in \mathcal{M}, \forall h \in \mathbb{N} \quad \mathbb{P}(M_{T+h} \in B \mid \mathcal{F}_T^M) = \mathbb{P}(M_{T+h} \in B \mid M_T). \quad (1.4)$$

For a countable space \mathbb{M} and a homogeneous chain $(M_n)_{n \in \mathbb{N}}$, we use the following notations:

$$\begin{aligned} &\forall (i, j) \in \mathbb{M}^2, n \in \mathbb{N}^*, \\ &\mathbb{P}(M_n = j \mid M_{n-1} = i) = p_{ij}. \end{aligned} \quad (1.5)$$

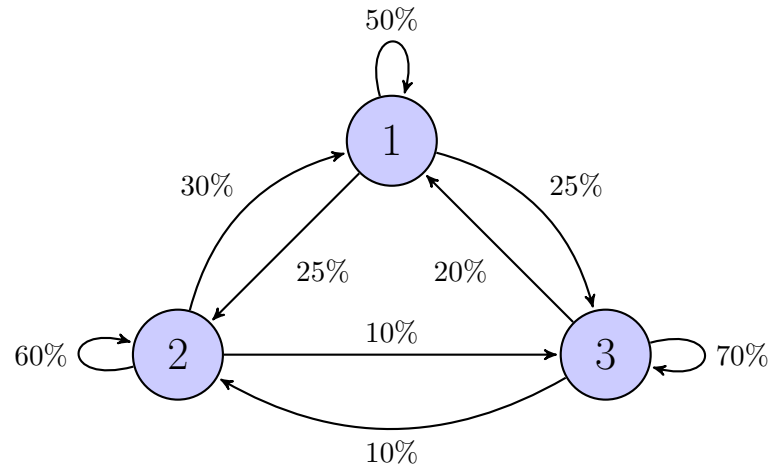


Figure 1.1 – Representation of a homogeneous Markov chain

with state space $\mathbb{M} = \{1, 2, 3\}$ and a one-step transition matrix $P = \begin{pmatrix} 0.50 & 0.25 & 0.25 \\ 0.30 & 0.60 & 0.10 \\ 0.20 & 0.10 & 0.70 \end{pmatrix}$.

Definition 5 (Transition matrix). Assuming the chain $(M_n)_{n \in \mathbb{N}}$ is homogeneous and \mathbb{M} is countable, the one-step transition matrix is defined by $P = (p_{ij})_{(i,j) \in \mathbb{M}^2}$. It is a stochastic matrix as $\forall i \in \mathbb{M}, \sum_{j \in \mathbb{M}} p_{ij} = 1$.

The k -step transition matrix is denoted $P^{(k)} = (p_{ij}^{(k)})_{(i,j) \in \mathbb{M}^2}$ and is such that $\forall n \in \mathbb{N}$:

$$p_{ij}^{(k)} = \mathbb{P}(M_{n+k} = j \mid M_n = i). \tag{1.6}$$

For instance, the figure 1.1 represents a Markov with three states and a one-step transition matrix, and a possible realization of this Markov chain is displayed in figure 1.2.

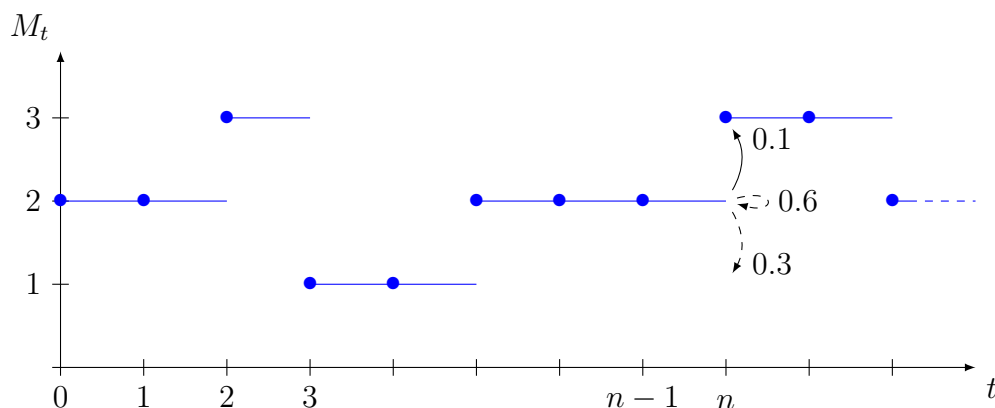


Figure 1.2 – Example of realization of a Markov chain with state space $\mathbb{M} = \{1, 2, 3\}$ and a one-step transition matrix

$$P = \begin{pmatrix} 0.50 & 0.25 & 0.25 \\ 0.30 & 0.60 & 0.10 \\ 0.20 & 0.10 & 0.70 \end{pmatrix}.$$

Property 1 (Chapman-Kolmogorov equations). The transition matrices satisfy the Chapman-Kolmogorov equations:

$$\forall s \leq k, \quad P^{(k)} = P^{(k-s)}P^{(s)} \quad (1.7)$$

and in fact one can show that $P^{(k)} = P^k$ and $P^{(0)} = I$.

1.1.2 Markov processes

Definition 6 (Stochastic process). Let $(A, \mathcal{A}, \mathbb{P})$ be a probability space and (E, \mathcal{E}) be a measurable space. A sequence $(Z_t)_{t \in \mathbb{R}^+}$ of random variables from $(A, \mathcal{A}, \mathbb{P})$ to (E, \mathcal{E}) is a *stochastic process*.

Definition 7 (Filtration). The σ -algebra generated by the sequences of random variables $(Z_t)_{t \leq v}$ is the *filtration* associated to the stochastic process $(Z_t)_{t \in \mathbb{R}^+}$, it is denoted by \mathcal{F}_v . The σ -algebra generated by $(Z_t)_{t < v}$ is denoted by \mathcal{F}_{v-} .

Definition 8 (Markov process, Homogeneity). The stochastic process $(Z_t)_{t \in \mathbb{R}^+}$ is a *Markov process* with values in (E, \mathcal{E}) if it verifies the memory-less property:

$$\forall B \in \mathcal{E}, \forall h > 0, t > 0, \quad \mathbb{P}(Z_{h+t} \in B \mid \mathcal{F}_t) = \mathbb{P}(Z_{h+t} \in B \mid Z_t). \quad (1.8)$$

The process is considered homogeneous when $\mathbb{P}(Z_{h+t} \in B \mid Z_t)$ does not depend on the time t .

Definition 9 (Stopping time). A random variable T with value in $(\mathbb{R}^+ \cup \{+\infty\}, \mathcal{B}(\mathbb{R}^+ \cup \{+\infty\}))$ is a *stopping time* for the process $(Z_t)_{t \in \mathbb{R}^+}$, if $\forall t, \{T \leq t\} \in \mathcal{F}_t$.

Definition 10 (σ -algebra on a stopping time). If T is a stopping time,

$$\mathcal{F}_T = \left\{ B \in \mathcal{F}_\infty = \sigma \left(\bigcup_{t \in \mathbb{R}^+} \mathcal{F}_t \right) \mid B \cap \{T \leq t\} \in \mathcal{F}_t, \forall t \geq 0 \right\} \quad (1.9)$$

Theorem 2 (Corollary of the Strong Markov property). A *Markov process* $(Z_t)_{t \in \mathbb{R}^+}$ verifies the strong Markov property and consequently for a finite stopping time T we have:

$$\forall B \in \mathcal{E}, \quad \mathbb{P}(Z_{T+h} \in B \mid \mathcal{F}_T) = \mathbb{P}(Z_{T+h} \in B \mid Z_T). \quad (1.10)$$

Definition 11 (Transition matrix). If the state space of the process (E, \mathcal{E}) is countable, and the process is homogeneous, one can define the t -step transition matrix. Let $p_{ij}(t) = \mathbb{P}(Z_{t+h} = j \mid Z_h = i) = \mathbb{P}(Z_t = j \mid Z_0 = i)$, the t -step transition matrix is then defined by

$$P(t) = \left(p_{ij}(t) \right)_{(i,j) \in E^2}. \quad (1.11)$$

It is a stochastic matrix as $\forall i \in E, \sum_{j \in E} p_{ij}(t) = 1$.

Property 2 (Chapman-Kolmogorov equations). Assuming the state space of the homogeneous Markov process (E, \mathcal{E}) is countable, the transition matrices satisfy the Chapman-Kolmogorov equations:

$$\forall s \leq t, \quad P(t) = P(t-s)P(s) \quad \text{with } P(0) = I \quad (1.12)$$

as

$$\forall (i, j) \in E^2, \quad \forall s \leq t, \quad p_{ij}(t) = \sum_{k \in E} p_{kj}(t-s)p_{ik}(s) \quad (1.13)$$

Sometimes it can be interesting to work with some local properties of the process. In order to describe the local evolution of a process one can use its infinitesimal generator. This quantity is defined as follows:

Definition 12 (Generator). We denote by $\mathcal{M}(E)$ the set of real-valued measurable functions on a set (E, \mathcal{E}) . For a homogeneous Markov process $(Z_t)_{t \in \mathbb{R}^+}$ with values in (E, \mathcal{E}) , let $D(\mathcal{Q}) \subset \mathcal{M}(E)$ be the set of bounded measurable functions such that for all $z \in E$ the following limit exists:

$$\lim_{t \downarrow 0} \frac{\mathbb{E}[f(Z_t) | Z_0 = z] - f(z)}{t}. \quad (1.14)$$

$D(\mathcal{Q})$ is called the domain of the generator \mathcal{Q} .

The infinitesimal generator is the linear operator \mathcal{Q} defined by

$$\mathcal{Q}f(Z_s) = \lim_{t \downarrow 0} \frac{\mathbb{E}[f(Z_{s+t}) | Z_s] - f(Z_s)}{t} \quad (1.15)$$

where $f \in D(\mathcal{Q})$.

A Markov chain as Markov process

Note that it is possible to extend any Markov chain M_n , evolving in a countable state space \mathbb{M} , to a Markov process (Z_t) , evolving in a state space $E = \mathbb{R} \times (-\infty, 1) \times \mathbb{M}$, by setting

$$Z_t = (t, t - \lfloor t \rfloor, M_{\lfloor t \rfloor}). \quad (1.16)$$

The coordinate $t - \lfloor t \rfloor$ is optional, it will be used in subsection 1.2.4 to express (Z_t) as a PDMP. Here we can check that (Z_t) is a Markov process as, $\forall t, u \in \mathbb{R}$, with $t > u$, it verifies:

$$\begin{aligned} \forall m \in \mathbb{M}, \quad \mathbb{P}(Z_t = (t, t - \lfloor t \rfloor, m) \mid \mathcal{F}_u) &= p_{M_u m}^{(\lfloor t \rfloor - \lfloor u \rfloor)} \\ &= \mathbb{P}(Z_t = (t, t - \lfloor t \rfloor, m) \mid Z_u = (u, u - \lfloor u \rfloor, M_u)). \end{aligned}$$

1.1.3 Poisson Processes

A Poisson process is frequently used as a model for counting events occurring one at a time, such as the number of births in a hospital, the number of arrivals at a service system, the number of accidents in a given section of a road, etc. In this section we present three equivalent definitions of the Poisson Processes. Then we introduce the inhomogeneous Poisson processes which are useful to model events having higher or lower chances of occurring in some periods of time.

Homogeneous Poisson processes

The Poisson process can first be defined as a process counting events separated by independent exponential times:

Definition 13 (Poisson process). We denote by $(S_n)_{n \in \mathbb{N}}$ a sequence of event times, with the convention that $S_0 = 0$, and we denote by $(T_n)_{n \in \mathbb{N}}$ the sequence of inter-event times such that $S_n = \sum_{k=0}^{n-1} T_k$. Let the number of events occurring in the time interval $[0; t]$ be

$$N_t = \max\{n \geq 0 \mid S_n \leq t\}. \quad (1.17)$$

The process $(N_t)_{t \geq 0}$ is said to be a Poisson process with parameter λ if the inter-event times are *iid* exponential random variables¹ of parameter λ .

An example of realization of a Poisson process is displayed in figure 1.3.

The second definition of Poisson processes can be given by setting the properties of its increments:

Definition 14 (Poisson process, properties of the increments). An integer valued stochastic process $(N_t)_{t \geq 0}$ is a Poisson process if and only if its increments verify the three following properties:

(i) Independent increments:

The increments over non-overlapping intervals are independent

(ii) Stationary increments:

The law of each increment over an interval $(s, t + s]$ is independent of s

(iii) Poisson distributed increments:

The increment over an interval of length t has a Poisson law of parameter λt :

$$\forall s, t > 0, \quad \mathbb{P}(N_{t+s} - N_s = k) = \frac{(\lambda t)^k}{k!} \exp[-\lambda t] \quad (1.18)$$

1. *iid* stands for independent and identically distributed

Note the independence of the increments implies the process has no memory. In particular the Poisson process is a Markov process, as:

$$\begin{aligned} \mathbb{P}(N_{t+s} = k \mid (N_k)_{k \leq s}) &= \mathbb{P}(N_{t+s} - N_s = k - N_s \mid (N_k)_{k \leq s}) \\ &= \mathbb{P}(N_{t+s} - N_s = k - N_s \mid N_s) \quad (\text{by independence of the increments}) \\ &= \mathbb{P}(N_{t+s} = k \mid N_s). \end{aligned}$$

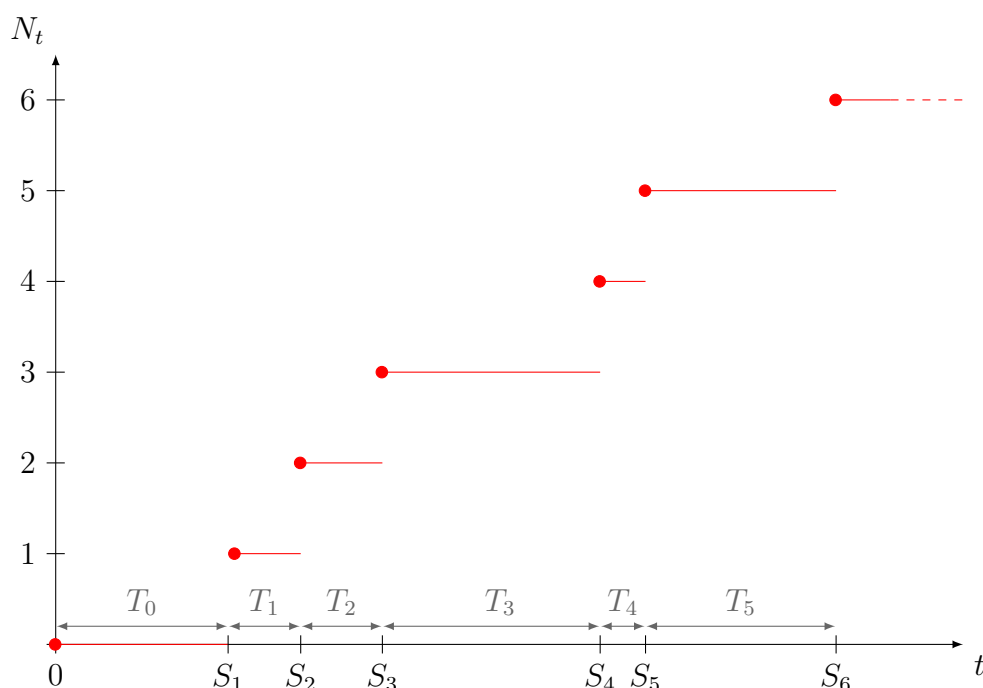


Figure 1.3 – Example of realization of a Poisson process

Thirdly, one can define a Poisson by its local behavior:

Definition 15 (Poisson process, local characterization). The integer valued stochastic process $(N_t)_{t \geq 0}$ is a Poisson process with parameter λ if and only if the following three conditions are satisfied:

- (i) The process $(N_t)_{t \geq 0}$ has independent increments
- (ii) $\mathbb{P}(N_0 = 0) = 1$,
- (iii) For any time $t \geq 0$, and for a time $h > 0$ (in the neighborhood of zero):

$$\mathbb{P}(N_{t+h} - N_t = 0) = 1 - \lambda h + o(h), \quad (1.19)$$

$$\mathbb{P}(N_{t+h} - N_t = 1) = \lambda h + o(h), \quad (1.20)$$

$$\mathbb{P}(N_{t+h} - N_t > 1) = o(h). \quad (1.21)$$

Proof that the definition 15 is equivalent to the definition 14. We only prove the implication "definition 15 \Rightarrow definition 14", as the other implication "definition 14 \Rightarrow definition 15" is straightforward. Assuming definition 15 we can obtain the derivative $\frac{d\mathbb{P}(N_t = k)}{dt}$. First note that for $k > 0$

$$\begin{aligned}
 \mathbb{P}(N_{t+h} = k) &= \mathbb{E}\left[\mathbb{P}(N_{t+h} = k \mid N_t)\right] \\
 &= \sum_{j=0}^k \mathbb{P}(N_{t+h} = k \mid N_t = j)\mathbb{P}(N_t = j) \\
 &= \sum_{j=0}^k \mathbb{P}(N_{t+h} - N_t = k - j \mid N_t = j)\mathbb{P}(N_t = j) \\
 &= \sum_{j=0}^k \mathbb{P}(N_h = k - j)\mathbb{P}(N_t = j) \quad \text{by independence of the increments} \\
 &= (1 - \lambda h + o(h))\mathbb{P}(N_t = k) + \lambda h\mathbb{P}(N_t = k - 1) + o(h). \tag{1.22}
 \end{aligned}$$

Therefore:

$$\frac{\mathbb{P}(N_{t+h} = k) - \mathbb{P}(N_t = k)}{h} = -\lambda\mathbb{P}(N_t = k) + \lambda\mathbb{P}(N_t = k - 1) + o(1). \tag{1.23}$$

Making h tend to 0, we get

$$\frac{d\mathbb{P}(N_t = k)}{dt} = \lambda\mathbb{P}(N_t = k - 1) - \lambda\mathbb{P}(N_t = k) \tag{1.24}$$

Proceeding in a similar fashion, for the case $k = 0$, we get

$$\frac{d\mathbb{P}(N_t = 0)}{dt} = \lambda\mathbb{P}(N_t = 0). \tag{1.25}$$

Using the initial condition $\mathbb{P}(N_0 = 0) = 1 \forall k \geq 1$, we can resolve the above equation, yielding

$$\mathbb{P}(N_t = 0) = \exp(-\lambda t). \tag{1.26}$$

Then using the initial condition $\mathbb{P}(N_0 = k) = 0$, we can solve (1.24) recursively to get

$$\mathbb{P}(N_t = k) = \frac{(\lambda t)^k}{k!} \exp[-\lambda t]. \tag{1.27}$$

This proves the statement (ii) and (iii) of the definition 14. \square

Inhomogeneous Poisson processes

In a Poisson process, the event rate is constant, but one might want to model events that occur at different rates according to the time period. For instance, if the events of interest are the arrivals of the consumers at a post office, the event rates should be higher

during rush hours, and lower during off-peak times. In order to model such situations, an Inhomogeneous Poisson process with a variable jump rate is better suited than a Poisson process which has a constant jump rate. One can define an Inhomogeneous Poisson process by its local behavior:

Definition 16 (Inhomogeneous Poisson process, local characterization). Let $\lambda : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ be a function that is Lebesgue-integrable on any finite interval $(a, b) \subset \mathbb{R}^+$. The integer valued stochastic process $(N_t)_{t \geq 0}$ is an Inhomogeneous Poisson process with a function of parameter λ if and only if:

- (i) The process $(N_t)_{t \geq 0}$ has independent increments
- (ii) $\mathbb{P}(N_0 = 0) = 1$
- (iii) For all time $\forall t > 0$, and for a time $h > 0$ (in the neighborhood of zero):

$$\mathbb{P}(N_{t+h} - N_t = 0) = 1 - \lambda(t)h + o(h), \quad (1.28)$$

$$\mathbb{P}(N_{t+h} - N_t = 1) = \lambda(t)h + o(h), \quad (1.29)$$

$$\mathbb{P}(N_{t+h} - N_t > 1) = o(h). \quad (1.30)$$

Following the same kind of reasoning as for the Poisson process, one can define the Inhomogeneous Poisson process by its increments.

Definition 17 (Inhomogeneous Poisson process, increments 'properties'). An integer valued stochastic process $(N_t)_{t \geq 0}$ is an inhomogeneous Poisson process if and only if its increments verify two following properties:

- (i) Independent increments:
The increments over non-overlapping intervals are independent
- (ii) The increments are distributed in the following manner:

$$\forall s, t > 0, \quad \mathbb{P}(N_{t+s} - N_s = k) = \frac{(\Delta_s(t))^k}{k!} \exp[-\Delta_s(t)] \quad (1.31)$$

where $\Delta_s(t) = \int_s^{t+s} \lambda(u) du$

1.1.4 Compound Poisson process

Compound Poisson processes can be viewed as a generalization of the Poisson process. The Poisson process assumes the events to arrive one at a time, this assumption is relaxed in the Compound Poisson processes. In practice, this process is used when we want to count events that arrive in batches, like the customers arriving at a restaurant, or the number of cars involved in an accident, demands occurring in batches, etc.

Let N_t be the number of batches arrived before time t , and let Y_n be the number of event contained in the $n - th$ batch. The process $(X_t)_{t > 0}$ counting the number of events

is defined by:

$$X_t = \sum_{i=1}^{N_t} Y_i. \quad (1.32)$$

Definition 18. The process $(X_t)_{t>0}$ is a compound Poisson process if and only if:

- (i) $(N_t)_{t \geq 0}$ is a Poisson process with rate λ (i.e; the batches arrive at constant rate)
- (ii) The sequence $(Y_n)_{n \in \mathbb{N}}$ is *iid* and independent of $(N_t)_{t \geq 0}$

Note that the compound Poisson process is more than just a counting process when it is defined in this manner. The compound Poisson process can also be used to model a quantity X_t impacted by events occurring at a rate λ . Indeed Y_n is not necessarily an integer or a positive random variable.

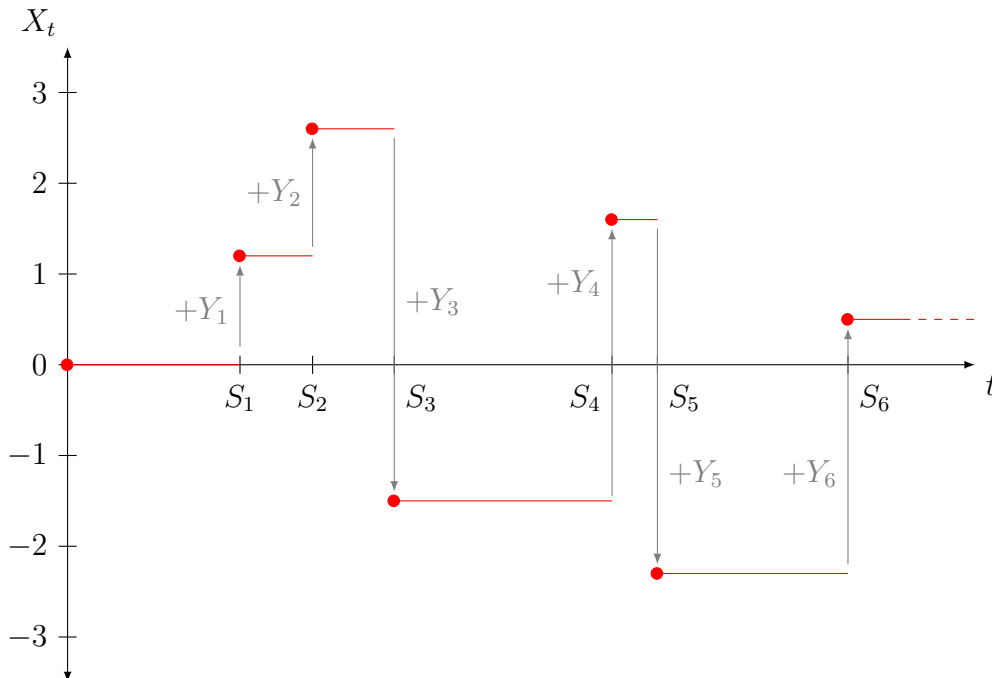


Figure 1.4 – Example of realization of a compound Poisson process

1.1.5 Continuous time Markov chains

A continuous time Markov chain (CTMC) can be viewed as a generalization of a Markov chain, where the time between jumps is an exponential random variable, instead of being fixed to 1.

Definition 19 (CTMC). The sequence of random variables $(M_t)_{t \in \mathbb{R}^+}$ is a CTMC in the countable measurable space $(\mathbb{M}, \mathcal{M})$, if and only if:

- (i) $(Y_n)_{n \in \mathbb{N}}$ is a homogeneous Markov chain with values in $(\mathbb{M}, \mathcal{M})$. We take the convention that this Markov Chain cannot jump on the current state, therefore its transition matrix $P = (p_{ij})_{(i,j) \in \mathbb{M}^2}$ is such that $\forall i \in \mathbb{M}, p_{ii} = 0$.

(ii) $(S_n)_{n \in \mathbb{N}}$ is a sequence of increasing random times such that $S_0 = 0$ and conditionally on $(Y_n)_{n \in \mathbb{N}}$, $T_n = S_{n+1} - S_n$ are independent exponential variables of parameter $\lambda_{Y_n} \in \mathbb{R}^+$.

(iii)

$$\forall t \in [S_n, S_{n+1}), \quad M_t = Y_n. \quad (1.33)$$

$(Y_n)_{n \in \mathbb{N}}$ is called the *embedded Markov chain* of the continuous time Markov chain $(M_t)_{t \in \mathbb{R}^+}$.

This definition implies that:

$$\begin{aligned} & \mathbb{P} \left(M_{S_n} = j, T_{n-1} \geq t \mid M_{S_{n-1}} = i, T_{n-2}, M_{S_{n-2}}, \dots, T_0, M_{S_0} \right) \\ &= \mathbb{P} \left(M_{S_n} = j, T_{n-1} \geq t \mid M_{S_{n-1}} = i, S_{n-1} \right) \\ &= \mathbb{P} \left(M_{S_1} = j, T_0 \geq t \mid M_0 = i, S_0 \right) \\ &= p_{ij} \exp[-\lambda_i t]. \end{aligned} \quad (1.34)$$

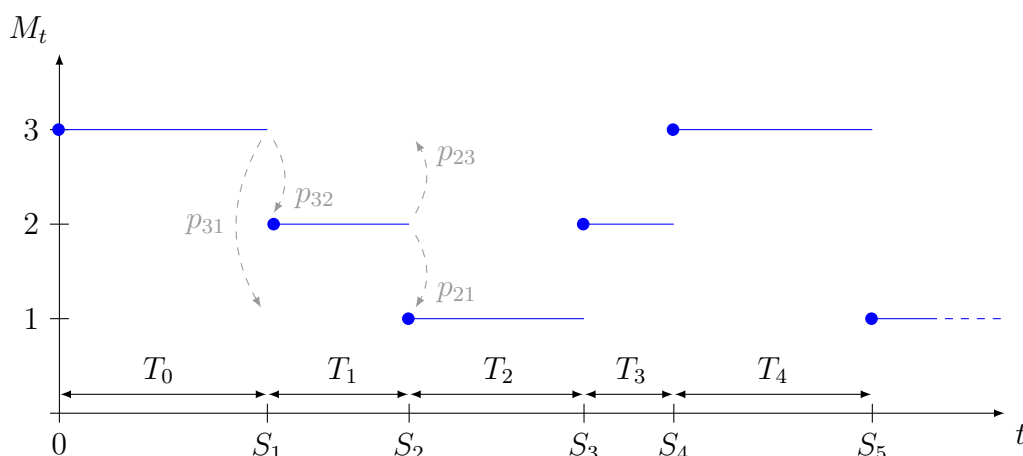


Figure 1.5 – Example of realization of a continuous time Markov chain with $\mathbb{M} = \{1, 2, 3\}$

Definition 20 (Generator, Forward and backward equations). The generator of the CTMC $(M_t)_{t \in \mathbb{R}^+}$ is the matrix $Q = D(P - I)$, where $D = \underset{i \in \mathbb{M}}{\text{diag}}(\lambda_i)$. The generator characterizes the local evolution of the chain, indeed denoting P_t the t -step transition matrix of $(M_t)_{t \in \mathbb{R}^+}$, we have:

$$\frac{dP_t}{dt} = QP_t \quad (\text{forward equation}) \quad (1.35)$$

$$\frac{dP_t}{dt} = P_t Q \quad (\text{backward equation}) \quad (1.36)$$

This implies that $\forall t \geq 0$, $P_t = P_0 e^{Qt}$. So $P_t = e^{Qt}$ as $P_0 = I$.

When they are coupled with the memory less property, these forward and backward equations fully define the distribution of the process $(M_t)_{t \in \mathbb{R}^+}$. Indeed it is possible to

provide an alternative definition to a continuous time Markov chain by specifying only its local behavior:

Definition 21 (CTMC local definition). A sequence of random variables $(M_t)_{t \in \mathbb{R}^+}$ is a continuous time Markov chain, if and only if the two following conditions are satisfied:

- (i) $\forall i, j \in \mathbb{M}, \quad \mathbb{P}(M_{h+t} = j \mid M_t = i) = \mathbb{1}_{i=j} + q_{ij}h + o(h)$
- (ii) knowing M_t , M_{t+h} is independent of the past values $(M_s)_{s < t}$.

1.2 Piecewise Deterministic Markov Processes

A PDMP is a process that follows a deterministic dynamic, but this dynamic can change at random or deterministic times. Such deterministic dynamics are defined by ordinary differential equations. Therefore this section starts with a few reminders about ordinary differential equations. Then we gradually present PDMPs, as we start to present a particular case of PDMP before presenting the full PDMP model. The particular case of PDMP we present is the PDMP without boundary. This model is the most common form of PDMP. The most general form of PDMPs being unsurprisingly PDMPs with boundaries. This distinction between PDMPs with and without boundary is important, indeed we will see that, contrarily to PDMPs without boundary, PDMPs with boundaries can be very singular processes.

1.2.1 Generalities on first order ordinary differential equations

Definition 22. An application $\mathbf{g} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is global *Lipschitz* if it exists a constant $C > 0$, such that we have:

$$\forall x_1 \in \mathbb{R}^d, \forall x_2 \in \mathbb{R}^d, \quad \left\| \mathbf{g}(x_1) - \mathbf{g}(x_2) \right\| \leq C \left\| x_1 - x_2 \right\|. \quad (1.37)$$

Corollary 1. The Cauchy-Lipschitz theorem or Picard-Lindelöf theorem implies that if \mathbf{g} is global Lipschitz, the differential equation

$$\frac{dX(t)}{dt} = \mathbf{g}(X(t)), \quad \text{such that } X(a) = x_a \in \mathbb{R}^d, \quad (1.38)$$

admits a unique global solution of class C^1 .

Let ϕ be the function on $\mathbb{R}^d \times \mathbb{R}$ such that $X(a+t) = \phi(x_a, t)$ is the solution to the differential equation. ϕ is called a *flow*.

One can deduce from the uniqueness of the solution that ϕ verifies the two following properties

Property 3. The application $\phi_t : x \rightarrow \phi(x, t)$ is invertible, and its inverse is continuous, so it is a homeomorphism. Indeed we have that $\phi_t^{-1}(x) = \phi(x, -t)$.

Property 4. The family $(\phi_t)_{t \in \mathbb{R}}$ is a group, meaning that for t and s in \mathbb{R} $\phi_{t+s} = \phi_t \circ \phi_s$, or more explicitly $\phi(x, t + s) = \phi(\phi(x, s), t)$.

Definition 23. If f is a \mathcal{C}^1 function, then

$$\frac{d}{dt} f(\phi(x, t)) = \sum_{i=1}^d \frac{\partial f}{\partial x_i}(\phi(x, t)) \mathbf{g}_i(\phi(x, t)), \quad (1.39)$$

where \mathbf{g}_i is the i -th coordinate of \mathbf{g}

Definition 24. We denote by \mathcal{X} the differential operator of the first order:

$$\mathcal{X}f(x) = \sum_{i=1}^d \frac{\partial f}{\partial x_i}(x) \mathbf{g}_i(x) \quad (1.40)$$

1.2.2 PDMPs without boundary

We start our presentation of PDMPs by introducing the class of PDMPs without boundary.

State space

A PDMP is a process that models the evolution of a hybrid variable Z_t . This variable is characterized as hybrid because it is made of two coordinates X_t and M_t of different natures: The first coordinate X_t being a continuous variable taking its values in \mathbb{R}^d , whereas the second coordinate is a discrete variable with values in a countable set \mathbb{M} . At a time $t \in \mathbb{R}$, the state of the PDMP is defined by:

$$Z_t = (X_t, M_t). \quad (1.41)$$

Throughout this thesis we call X_t the position of the PDMP and M_t the mode of the PDMP. The state space of the PDMP is denoted E , and for each mode $m \in \mathbb{M}$ we denote E_m the set of all the states with mode m . For a PDMP without boundary $E_m = \{(x, m), x \in \mathbb{R}^d\}$, and the state space is defined by:

$$E = \bigcup_{m \in \mathbb{M}} E_m = \mathbb{R}^d \times \mathbb{M} \quad (1.42)$$

One can equip E with the metric $dist$ defined by:

$$\forall z_1 = (x_1, m_1), z_2 = (x_2, m_2) \in E, \quad (1.43)$$

$$dist(z_1, z_2) = \begin{cases} 1 & \text{if } m_1 \neq m_2 \\ \frac{2}{\pi} \tan^{-1}(\|x_1 - x_2\|) & \text{if } m_1 = m_2 \end{cases}. \quad (1.44)$$

By taking the topology induced by this metric, we can define a Borelian σ -algebra on E . We denoted this Borelian σ -algebra on E by $\mathcal{B}(E)$.

Deterministic dynamics

Within a mode m , the position follows a differential equation. For each $m \in \mathbb{M}$, we let $F_m : \mathbb{R}^d \rightarrow \mathbb{R}^d$ be a function, which is global Lipschitz. Assuming $M_t = m$ for $t \in [s, s+T)$, then X_t satisfies:

$$\frac{dX_t}{dt} = F_m(X_t). \quad (1.45)$$

The figure 1.6 shows a representation of the state space. Each plan represents a subset E_m , with the vector fields induced by the associated differential equations (1.45). Note the equation (1.45) also allows that the function F_m can also depend on t , because we can include t in the coordinates of X_t .

The flow function solution of the differential equation (1.45) is denoted $\phi_m(x, \cdot)$, so we have

$$\begin{aligned} \forall t \in [s, s+T), \\ X_{t+s} = \phi_m(X_s, t). \end{aligned} \quad (1.46)$$

We also define the associated flow function Φ for the state, such that:

$$\begin{aligned} \forall s \in [0, T), \\ Z_{t+s} = \Phi_{Z_t}(s) = (\phi_{M_t}(X_t, s), M_t) = (X_{t+s}, M_t) \end{aligned} \quad (1.47)$$

Z_t follows this flow function until it reaches a point of discontinuity in its value. Such discontinuity is called a *jump*. Between two jumps or discontinuities, the trajectory of Z_t is determined by the flow Φ_{Z_t} hence the designation of "piecewise deterministic" processes. We denote S_k the time of the k -th jump of the process, and $T_k = S_{k+1} - S_k$ the time between two consecutive jumps. The figure 1.8 shows an example of trajectory associated with the vector field of figure 1.6.

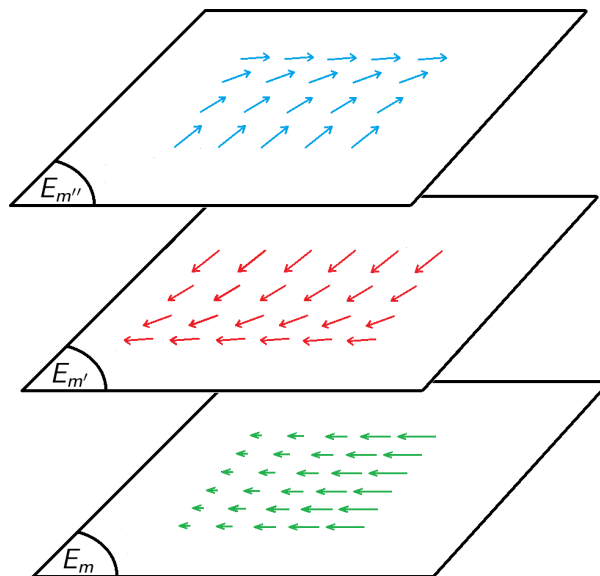


Figure 1.6 – Schematic representation of the state space of a PDMP without boundary

Timing of the jumps

Knowing $Z_t = z = (x, m)$, let T be the time until the next jump. This time T is random. Its law is expressed through a state-related intensity function $\lambda : E \rightarrow \mathbb{R}^+$. We have:

$$\mathbb{P}(T \leq h \mid Z_t = z) = 1 - \exp \left[- \int_0^h \lambda(\Phi_z(u)) du \right]. \quad (1.48)$$

One can also provide a time-related intensity by taking $\lambda_z : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that $\lambda_z(u) = \lambda(\Phi_z(u))$. We will assume that $\lim_{t \rightarrow \infty} \int_0^t \lambda_z(u) du = +\infty$. Using the following notation $\Lambda_z(h) = \int_0^h \lambda_z(u) du$, the law of T , knowing $Z_t = z$ can be expressed through λ_z with:

$$\mathbb{P}(T \leq h \mid Z_t = z) = 1 - e^{-\Lambda_z(h)} \quad (1.49)$$

$$= \int_0^h \lambda_z(u) e^{-\Lambda_z(u)} du. \quad (1.50)$$

Note that any continuous law with positive support can be specified through an intensity. If f_z is the density of T knowing $Z_t = z$, we have

$$\forall u > 0, \quad \lambda_z(u) = \frac{f_z(u)}{1 - \int_0^u f_z(v) dv} \quad \text{and} \quad f_z(u) = \lambda_z(u) e^{-\Lambda_z(u)}. \quad (1.51)$$

Arrival state of a jump

Definition 25 (Markovian kernel). Let (A, \mathcal{A}) and (E, \mathcal{E}) be two measurable spaces. An application $Q : A \times \mathcal{E} \rightarrow [0, 1]$ is a *Markovian kernel* from (A, \mathcal{A}) to (E, \mathcal{E}) if $\forall z \in A$, $Q(z, \cdot)$ is a probability on (E, \mathcal{E}) , and $\forall B \in \mathcal{E}$, $z \rightarrow Q(z, B)$ is \mathcal{A} -measurable.

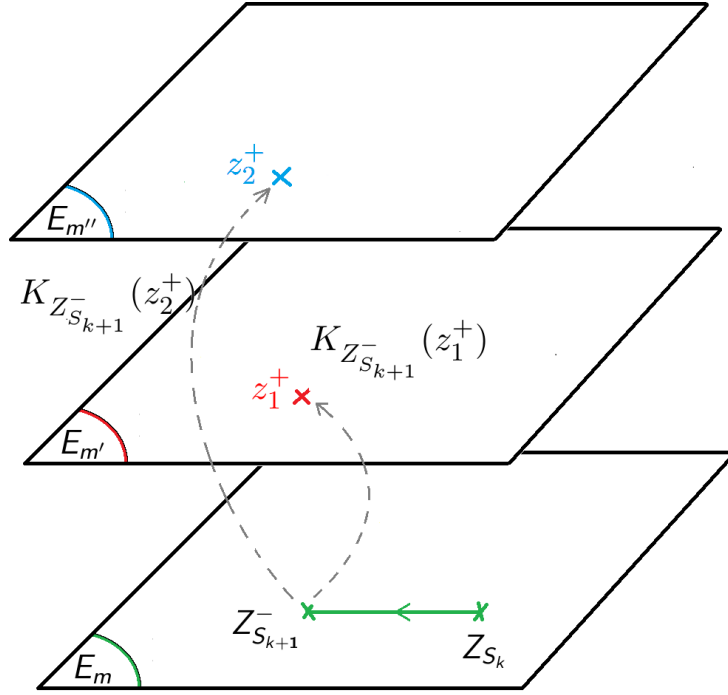


Figure 1.7 – Some possible arrival states of a jump in a PDMP without boundary

Let $\mathcal{K} : (E, \mathcal{B}(E)) \rightarrow [0, 1]$ be a Markovian kernel from $(E, \mathcal{B}(E))$ to $([0, 1], \mathcal{B}([0, 1]))$. To ease the presentation we use the notation $\mathcal{K}_z(\cdot)$ instead of $\mathcal{K}(z, \cdot)$, and we denote $K_z(\cdot)$ the density of the probability \mathcal{K}_z with respect to a reference measure ν_z :

$$\forall B \in \mathcal{B}(E), \quad \mathcal{K}_z(B) = \int_B K_z(z^+) d\nu_z(z^+), \quad (1.52)$$

$$\text{and for } h \in \mathcal{M}(E), \text{ such that } h \text{ bounded,} \quad \mathcal{K}_z(h) = \int_E h(z^+) K_z(z^+) d\nu_z(z^+). \quad (1.53)$$

Knowing Z_t , we have a jump at time $t+T$. Right before this time, one considers the process is in the state $Z_{t+T}^- = \lim_{s \uparrow T} \Phi_{Z_t}(s) = z^-$. We consider that the state Z_{t+T}^- is the departure state of the jump. The arrival state of the jump, which is Z_{t+T} , is chosen randomly. Its law is given by the probability $\mathcal{K}_{Z_{t+T}^-}$ such that:

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}(Z_{t+T} \in B \mid Z_{t+T}^- = z^-) = \mathcal{K}_{z^-}(B). \quad (1.54)$$

As a jump implies to jump to an other state, when ν_z has discrete parts the jump kernel must satisfy:

$$K_z(z) = 0, \quad \forall z \in E. \quad (1.55)$$

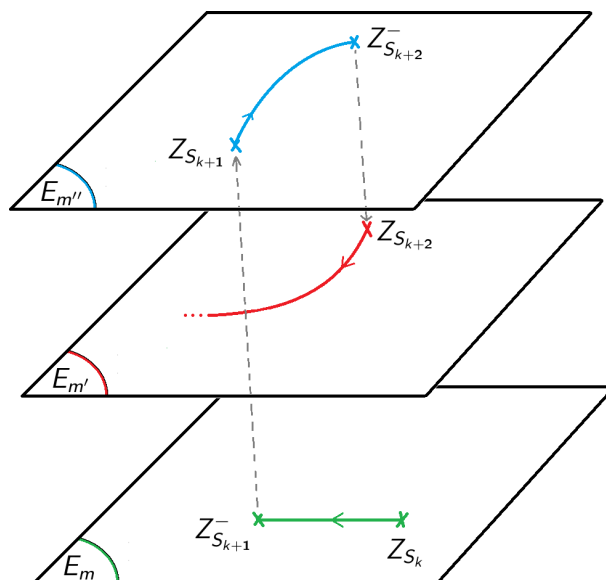


Figure 1.8 – Example of a trajectory of a PDMP without boundary

Generate a trajectory

As advised in [19, 18, 27], in order to generate a realization of the PDMP on a time interval $[0, t_f]$, one follows these steps:

1. Start at a state $Z_0 = z_0$
2. Generate T the time of the next jump using (1.48)
3. Follow the flow Φ until T using (1.47)
4. Generate $Z_T = z_T$ the arrival state of the jump knowing the departure state is $Z_T^- = \Phi_z(T)$ using (1.54)
5. Starting with z_T , repeat steps 1 to 4 until a trajectory of size t_f is obtained

Note we are sure to reach T_f in a finite number of jumps, because the intensities are integrable.

Homogeneity

Note that with the definition we used a PDMP is a homogeneous process, indeed the four elements characterizing a PDMP: E , Φ , λ , and K do not depend on the time. But this definition also covers the non-homogeneous case: indeed, it is possible to add a dependency on the time by simply adding the time t in the coordinates of the position X_t .

Example

A PDMP without boundary can model a differential equation with random coefficients. For instance $\mathbb{M} = \{-1, 0, 1\}$, $\frac{dX_t}{dt} = M_t$ and $\lambda(z) = 2$ for $z = (x, m) \in E$, and

$\mathcal{K}_z(\{(x, m)\}) = \frac{1}{2}\mathbb{1}_{x=x^-}\mathbb{1}_{m \neq m^-}$. In this case M_t is independent of the values of X_t , because λ depends only on the modes, and because M_S , the arrival mode of a jump at a time S , only depends on the previous mode M_S^- . M_t is therefore a continuous time Markov chain, which values determine the coefficient of the differential equation on X_t . An example of trajectory for this PDMP is presented in figure 1.9.

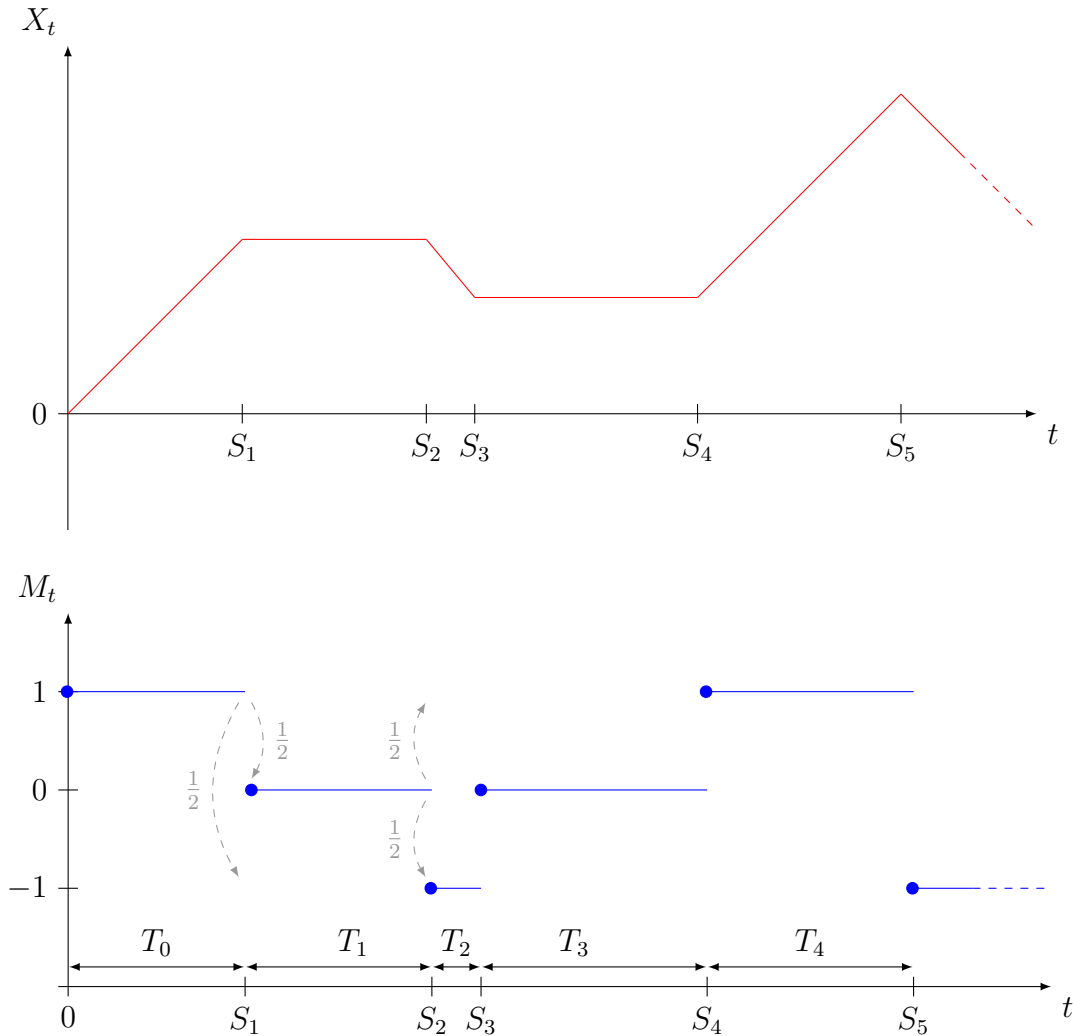


Figure 1.9 – Example of realization of the position and mode of a PDMP without boundary. This PDMP being defined by $\mathbb{M} = \{-1, 0, 1\}$, $\frac{dX_t}{dt} = M_t$, $\lambda(z) = 2$, and $\mathcal{K}_z(\{(x, m)\}) = \frac{1}{2}\mathbb{1}_{x=x^-}\mathbb{1}_{m \neq m^-}$

Note the position is not necessarily continuous. For instance, one can change the kernel so that the position is reset to zero at each jump. Such PDMP would yield realization similar to the one displayed in Figure 1.10.

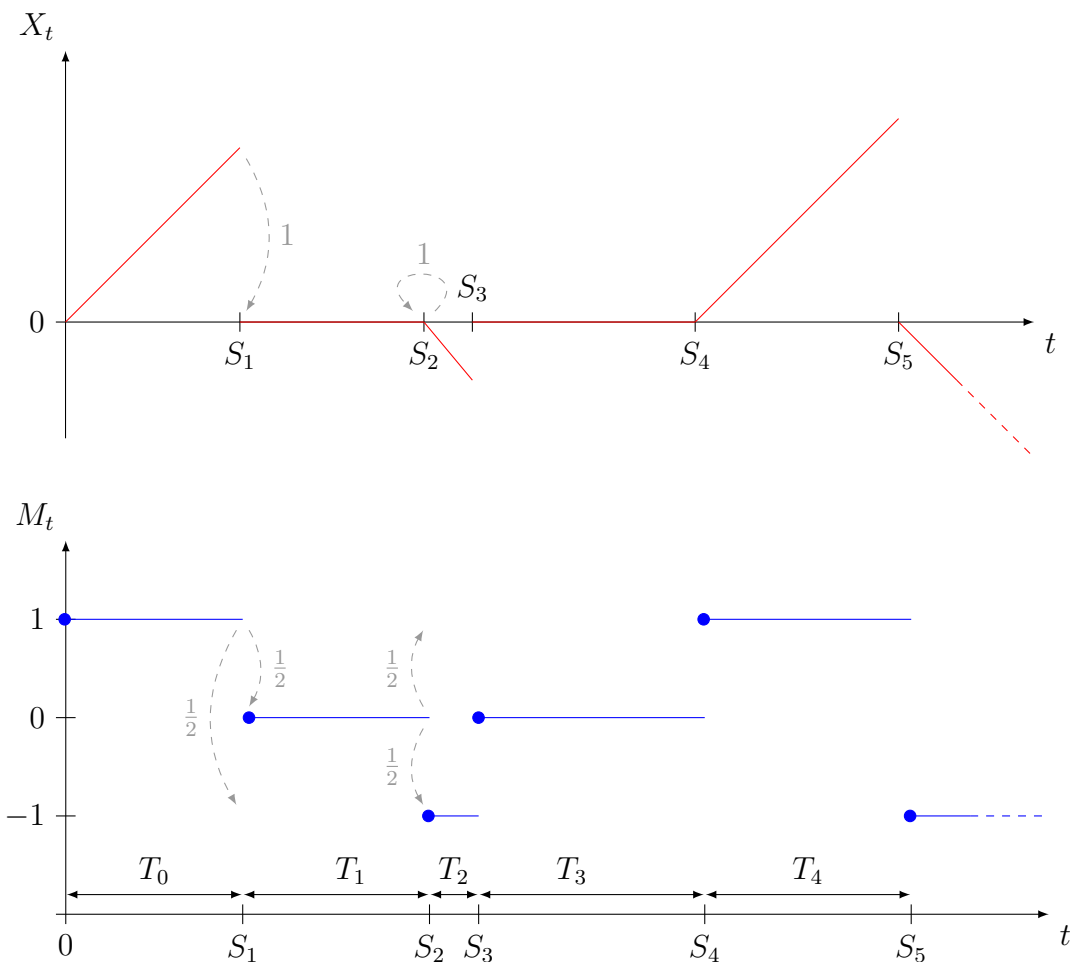


Figure 1.10 – Example of realization of the position and mode of a PDMP without boundary. This PDMP being defined by $\mathbb{M} = \{-1, 0, 1\}$, $\frac{dX_t}{dt} = M_t$, $\lambda(z) = 2$, and $\mathcal{K}_{z^-}(\{(x, m)\}) = \frac{1}{2} \mathbb{1}_{x=0} \mathbb{1}_{m \neq m^-}$

1.2.3 PDMPs with boundaries

Restricted state space

In a PDMP with boundaries the state space is different. Indeed for each mode m , one can restrict the position to an open set $\Omega_m \subset \mathbb{R}^d$. See Figure 1.11. Therefore the set of states with mode m becomes $E_m = \{(x, m), x \in \Omega_m\} = \Omega_m \times \{m\}$ and the state space becomes the open set:

$$E = \bigcup_{m \in \mathbb{M}} E_m = \bigcup_{m \in \mathbb{M}} \{(x, m), x \in \Omega_m\}. \quad (1.56)$$

Deterministic dynamics

Within the restricted set of positions Ω_m the dynamic of the position is still given through a differential equation like in (1.45), and the flow function Φ still gives the evolu-

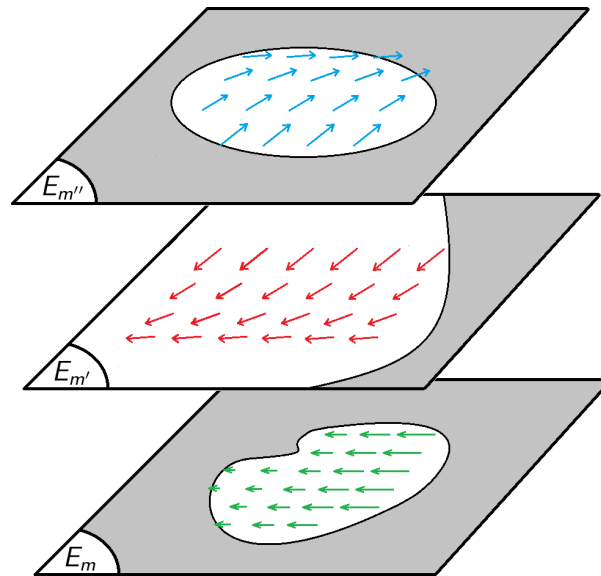


Figure 1.11 – Representation of the state space of a PDMP with boundaries

tion of the state until the next jump. See Figure 1.11. The difference is that now, when the flow brings the state to the boundary of E_m , the process has to jump to another location to stay in the state space, as shown in Figure 1.12

We denote the boundary of E_m by ∂E_m . The time until we hit a boundary starting from a state $z = (x, m)$ is denoted t_z^* , it is defined by $t_z^* = \inf\{t > 0, \Phi_z(t) \in \partial E_m\}$.

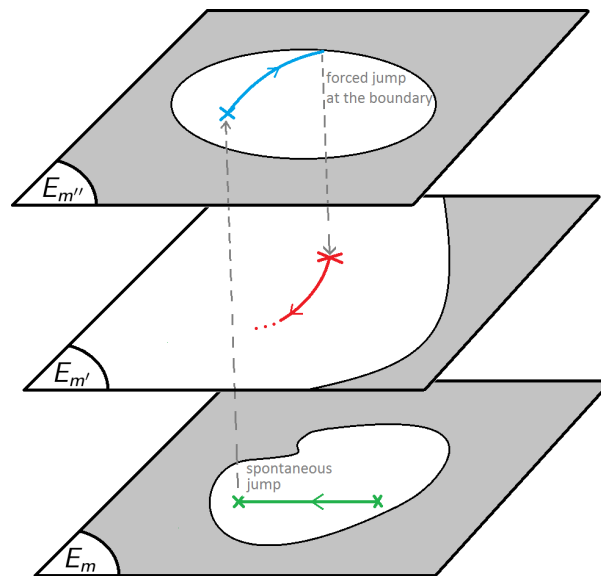


Figure 1.12 – Representation of a trajectory of a PDMP with boundaries

Timing of the jumps

As the state space is now restricted the law of the timing of the jump is different. Assume $Z_t = z = (x, m)$, and that we want to characterize the law of T , the time until the next jump after time t . As a jump is triggered if we hit the boundary ∂E_m , the law of T becomes:

$$\mathbb{P}(T \leq h \mid Z_t = z) = \begin{cases} 1 - \exp\left[-\int_0^h \lambda(\Phi_z(u)) du\right] & \text{if } h < t_z^*, \\ 1 & \text{if } h \geq t_z^*. \end{cases} \quad (1.57)$$

Written with the time-related intensity $\lambda_z = \lambda \circ \Phi_z$, we have:

$$\mathbb{P}(T \leq h \mid Z_t = z) = \begin{cases} 1 - \exp\left[-\int_0^h \lambda_z(u) du\right] & \text{if } h < t_z^*, \\ 1 & \text{if } h \geq t_z^*. \end{cases} \quad (1.58)$$

An example of the cumulative distribution function of T is displayed in figure 1.13

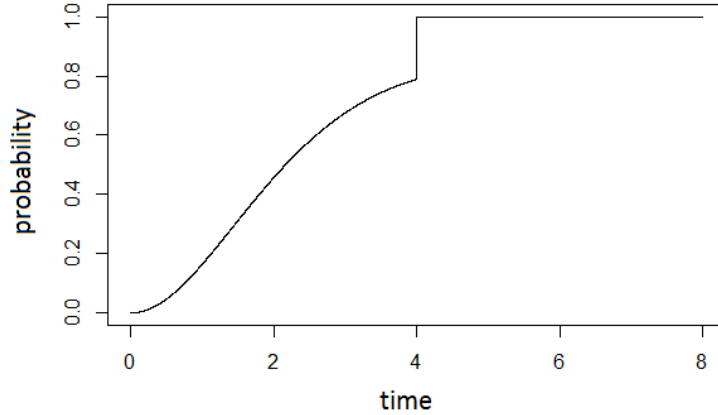


Figure 1.13 – An example of the cumulative distribution function of T , where $x \in \mathbb{R}^+$, $z = (x, m)$, $\Phi_z(t) = (x + t, m)$, $\lambda(z) = \frac{x(5-x)}{12}$, and $t_z^* = 4$

Destination of the jumps

The departure state of a jump Z_{t+T}^- now belongs to \bar{E} (the closure of E). The probability $\mathcal{K}_{Z_{t+T}^-}$ still takes values within the open measurable set $(E, \mathcal{B}(E))$. The law of the arrival state of a jump is still defined through equations (1.54) and (1.52), and jumps onto the departure state are still forbidden. $\forall z^- \in E$, If ν_{z^-} has a Dirac component on z^- :

$$K_{z^-}(z^-) = 0. \quad (1.59)$$

Similarly if the departure state z^- is on the boundary ∂E , the jump cannot land on the departure state z^- as \mathcal{K}_{z^-} is defined only on the open set E which does not contain z^- .

Generate a trajectory

In order to generate a realization of a PDMP with boundaries on a time interval $[0, t_f]$, one can proceed like before, except the timing of jump (step 2), is now affected by the boundaries (equation (1.58)):

1. Start at a state $Z_0 = z_0$
2. Generate T the time of the next jump using (1.58)
3. Follow the flow Φ until T using (1.47)
4. Generate $Z_T = z_T$ the arrival state of the jump knowing the departure state is $Z_T^- = \Phi_z(T)$ using (1.54)
5. Starting with z_T , repeat steps 1 to 4 until a trajectory of size t_f is obtained

Example of PDMP with boundaries

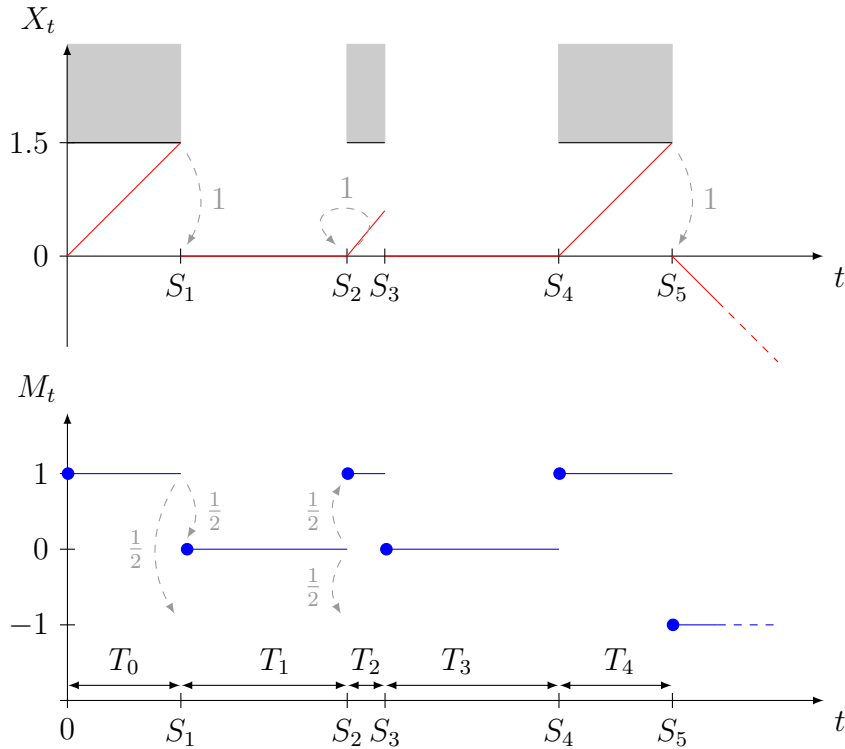


Figure 1.14 – Example of realization of the position and mode of a PDMP without boundary. This PDMP being defined by $\mathbb{M} = \{1, 0, -1\}$, $\frac{dX_t}{dt} = M_t$, $\lambda(z) = 2$, and $\mathcal{K}_{z^-}((x, m)) = \frac{1}{2} \mathbb{1}_{x=0} \mathbb{1}_{m \neq m^-}$, and $E_1 = \{(x, 1) | x \in (-\infty, 1.5)\}$, $E_0 = \{(x, 0) | x \in \mathbb{R}\}$, $E_{-1} = \{(x, -1) | x \in \mathbb{R}\}$

An example of PDMD with boundaries can be given by adding a boundary of the PDMP displayed previously in Figure 1.10. Within the mode $m = 1$, we set $\Omega_1 = (-\infty, 1.5)$ and therefore $E_1 = \{(x, 0) | x \in (-\infty, 1.5)\}$ and we let the other character-

istics $(\Phi, \lambda, \mathcal{K})$ and $E_0 = \{(x, 0) | x \in \mathbb{R}\}$, and $E_{-1} = \{(x, -1) | x \in \mathbb{R}\}$ unchanged. A realization of this PDMP with a boundary is displayed in Figure 1.14.

Trajectories of PDMP

$\mathbf{Z} = (Z_t)_{t \in [0, t_f]}$ denotes a trajectory defined on the interval $[0, t_f]$. For trajectories defined on an interval $[0, s]$ starting at 0, we precise the upperbound of the interval in the index, so that $\mathbf{Z}_s = (Z_t)_{t \in [0, s]}$. If the interval of definition of the trajectory is right-open then we add a "-" to the index, so that $\mathbf{Z}_{s-} = (Z_t)_{t \in [0, s)}$. For trajectories defined on a different interval I , we precise the interval on definition in index, so that $\mathbf{Z}_I = (Z_t)_{t \in I}$. We also denote by $n(\mathbf{Z}_I)$ the number of jumps in a trajectory \mathbf{Z}_I . For a trajectory \mathbf{Z}_I , we denote S_k the time of the k -th jump, and the time between the k -th and the $k + 1$ -th jumps is T_k . If a is the lowerbound of I and b is its upperbound: we take the convention that $S_0 = a$, and $S_{n(\mathbf{Z}_I)+1} = b$ and $T_{n(\mathbf{Z}_I)+1} = b - S_{n(\mathbf{Z}_I)}$.

As a trajectory of a PDMP follows a deterministic flow piecewisely, it is possible to sum up the random trajectory \mathbf{Z} by only keeping the information on its jumps. By the information on its jumps we mean the timing between two consecutive jumps in \mathbf{Z} and the arrival of each jump in \mathbf{Z} . The vector gathering this information is called the skeleton of the trajectory, it is defined by: $(Z_{S_0}, T_1, Z_{S_1}, \dots, T_{n(\mathbf{Z})}, Z_{S_{n(\mathbf{Z})}}, T_{n(\mathbf{Z})+1})$.

There is a one to one connection between the set of skeletons and the set of trajectories satisfying (1.47). Indeed each trajectory has a skeleton, and from any skeleton we can build a trajectory by filling the gaps between the jump with flow Φ via equation (1.47). For a time $s > 0$, we denote by Θ_s the map that changes a trajectory defined on an interval $[0, s]$ in its skeleton:

$$\Theta_s : \mathbf{Z}_s \longrightarrow (Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_s)}. \quad (1.60)$$

Denoting by \mathbf{E}_s the space of the trajectories of the PDMP (Z_t) defined on $[0, s]$, we have:

$$\mathbf{E}_s = \Theta_s^{-1} \left(\bigcup_{n \in \mathbb{N}^*} \left\{ (z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n \mid \sum_{i=0}^n t_i = s, \forall k < n, K_{\Phi_{z_{s_k}}(t_k)}(z_{s_k}) > 0 \right\} \right). \quad (1.61)$$

Similarly for any time $s > 0$, we denote by Θ_{s-} the map that changes a trajectory defined on an interval $[0, s)$ in its skeleton:

$$\Theta_{s-} : \mathbf{Z}_{s-} \longrightarrow (Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{s-})}. \quad (1.62)$$

we denote by \mathbf{E}_{s-} the space of the trajectories of the PDMP (Z_t) defined on $[0, s)$, and

we have:

$$\mathbf{E}_{s^-} = \Theta_{s^-}^{-1} \left(\bigcup_{n \in \mathbb{N}^*} \left\{ (z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n \mid \sum_{i=0}^n t_i = s, t_n > 0, \forall k < n, K_{\Phi_{z_{s_k}}(t_k)}(z_{s_k}) > 0 \right\} \right). \quad (1.63)$$

Infinitesimal generator

Remember we denote by \mathcal{X} the differential operator of the first order such that for a function $f : E \rightarrow \mathbb{R}$:

$$\mathcal{X}f(x) = \sum_{i=1}^d \frac{\partial f}{\partial x_i}(x) \mathbf{g}_i(x) \quad (1.64)$$

where \mathbf{g}_i is the i -th coordinate of \mathbf{g} when $\mathbf{g}(X_t) = \frac{\partial X_t}{\partial t}$.

Definition 26. For a function f which is integrable along the flow, i.e. such that $t \rightarrow f(\Phi_z(t))$ is absolutely continuous, the infinitesimal generator of a PDMP is the linear operator defined by:

$$\forall z \in E, \quad \mathcal{Q}f(z) = \mathcal{X}f(z) + \lambda(z) \left(\int_{z^+ \in E} K_z(z^+) f(z^+) d\nu_z(z^+) - f(z) \right) \quad (1.65)$$

Working on PDMP with boundaries and the issue of the topology

PDMP with boundaries are complex processes, and working with these processes can be challenging. There are three main points of difficulty to keep in mind when we work with these processes. Firstly, a PDMP models a hybrid variable that has continuous coordinates and discrete coordinates, therefore, the state space $E = \bigcup_{n \in \mathbb{M}} E_n$ is by essence discontinuous and is not Euclidean. Because of the discrete coordinates and the shapes of the open sets Ω_n , we have no obvious metric on the state space. In order to work with an easier topology, one option is to avoid working directly with the states, by working with their image through a real function defined on the state space. Secondly, the inter-jump times are hybrid random variables whose distributions have continuous and discrete parts, which can be tricky to manipulate. We will see in part II that the PDMPs are very degenerate processes because of these hybrid random variables involved in their laws. Thirdly, the trajectories on an interval of time are complex objects, not only because they evolve in the space E which does not have a metric, but also because their skeletons have variable sizes. Therefore it is also difficult to define a metric on the space of trajectories.

1.2.4 Markovian object as PDMPs

We end this subsection on PDMP by reviewing how classical Markovian objects can be modeled by a PDMP.

Markov chain as a PDMP

We have seen that a Markov chain $(M'_n)_{n \in \mathbb{N}}$ can be extended into a Markov process $(Z_t)_{t \geq 0}$ by setting

$$Z_t = (t, t - \lfloor t \rfloor, M'_{\lfloor t \rfloor}). \quad (1.66)$$

This process can be seen as a PDMP defined by

— A position $X_t = (t, t - \lfloor t \rfloor)$ and a mode $M_t = M'_{\lfloor t \rfloor}$.

— a differential equation $\frac{dX_t}{dt} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

— In order to force jumps when t is an integer we put a boundary on the second coordinate of X_t , setting: $\forall m \in \mathbb{M}$, $\Omega_m = (-\infty, +\infty) \times (-\infty, 1)$. Therefore we have a state space defined by:

$$E = \bigcup_{m \in \mathbb{M}} E_m = \bigcup_{m \in \mathbb{M}} \{(x_1, x_2, m) \mid x_1 \in \mathbb{R}, x_2 \in (-\infty, 1)\} \quad (1.67)$$

— A null jump intensity: $\lambda = 0$

— A jump kernel corresponding to the kernel of the Markov chain:

$$\forall u \in \mathbb{N}, \mathcal{K}_{(u,1,i)}(\{(u, 0, j)\}) = p_{ij}(u), \quad (1.68)$$

where $p_{ij}(u)$ is the (i, j) coefficient of the one-step transition matrix of the Markov chain $(M'_n)_{n \in \mathbb{N}}$ at time $u \in \mathbb{N}$.

Poisson process as a PDMP

A Poisson process $(N'_t)_{t \geq 0}$ with jump variable intensity $\lambda'(t)$ can be expressed as a PDMP $(Z_t)_{t \geq 0}$ by setting

$$Z_t = (t, N'_t, 0). \quad (1.69)$$

This PDMP would be defined by

— A position $X_t = (t, N'_t) \in \mathbb{R} \times \mathbb{N}$ and a simple set of mode: $\mathbb{M} = \{0\}$, so that $\forall t, M_t = 0$.

This gives a state space $E = \mathbb{R}^2 \times \{0\}$

— A differential equation $\frac{dX_t}{dt} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

— A jump intensity:

$$\forall t \in \mathbb{R}, n \in \mathbb{N}, \quad \lambda(t, n, 0) = \lambda'(t) \quad (1.70)$$

— A jump kernel:

$$\forall t \in \mathbb{R}, n \in \mathbb{N}, \quad \mathcal{K}_{(t,n,0)}(\{(t, n+1, 0)\}) = 1 \quad (1.71)$$

Compound Poisson process as a PDMP

A Compound Poisson process (X_t) is defined by

$$X_t = \sum_{i=1}^{N_t} Y_i, \quad (1.72)$$

where (N_t) is a Poisson process of intensity $\lambda_0 > 0$ and the Y_i are independent random variables, can be expressed as in a PDMP (Z_t) by setting

$$Z_t = (X_t, N_t, 0). \quad (1.73)$$

We denote by f_{Y_i} the density of Y_i with respect to a reference measure ν_i on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. We assume that $\mathbb{P}(Y_i = 0) = 0$. The characteristics of this PDMP would be:

- A set space $E = \mathbb{R}^2 \times \{0\}$
- A differential equation $\frac{dX_t}{dt} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- A jump intensity:

$$\forall x, n \in \mathbb{R}, \quad \lambda(x, n, 0) = \lambda_0 \quad (1.74)$$

- A jump kernel: $\forall x, n \in \mathbb{R}, \nu_{(x,n,0)} = \nu_n$ and

$$\mathcal{K}_{(x,n,0)}\left(\{(x+A) \times \{n+1, 0\}\}\right) = \int_A f_{Y_{n+1}}(y) d\nu_{(x,n,0)}(y) \quad (1.75)$$

Continuous time Markov chain as a PDMP

A homogeneous continuous time Markov chain $(M'_t)_{t \geq 0}$ with jump intensities $(\lambda'_m)_{m \in \mathbb{M}}$ and transition $P = (p_{ij})_{i,j \in \mathbb{M}}$, can be expressed as a PDMP $(Z_t)_{t \geq 0}$ by setting

$$Z_t = (0, M'_t). \quad (1.76)$$

The characteristics of this PDMP would be:

- A set space $E = \mathbb{R} \times \mathbb{M}$
- A differential equation $\frac{dX_t}{dt} = 0$
- A jump intensity:

$$\forall x, n \in \mathbb{R}, \quad \lambda(0, m) = \lambda'_m \quad (1.77)$$

- A jump kernel:

$$\forall x, n \in \mathbb{R}, \quad \mathcal{K}_{(0,i)}\left(\{(0, j)\}\right) = p_{ij} \quad (1.78)$$

1.3 Model a multi-component system with a PDMP

Remember that, when they were first introduced by M.H.A Davis [18], PDMPs were meant to include all the Markovian objects that do not include diffusion. Consequently, PDMPs benefit from high modeling capacities, and most of the systems used for power generation can be modeled by using a PDMP. Such PDMP models have been proposed by several authors already [54, 27]. In this section we present our model of the state of a power generation system.

1.3.1 Model the evolution of the state of the system

We want to eventually assess the probability of failure of a system used for power generation. Such systems fail when the physical variables characterizing the system enter a critical region. For instance, one can consider a dam as failing when the water level inside the dam exceeds a security threshold, or if the dam is overflowed. Similarly, a nuclear power-plant fails when the values of the pressure and temperature of its core enter the liquid phase and lead to the fusion of the core.

Consequently, we need an accurate model of the physical variables. The evolution of such physical variables is ruled by differential equations issued from the laws of physics. But these differential equations change according to the statuses of the components within the system. Indeed, the physics inputs, and so the differential equations, change when some components are activated, or deactivated, or when some components fail. For instance the physical variables can enter the critical region corresponding to the system failure, only if a sufficient number of the basic components of the system are damaged.

The values of the physical variables also influence back the statuses of the components in two different ways. Firstly because in industrial systems there are some automatic control mechanisms, as some components are programmed to turn on or off when the physical variables cross some thresholds. Secondly because the values of the physical variables can impact the failure rates of the components. This is often the case when the physical variables include the pressure on a component: the more pressure there is on the component the more likely it is to fail.

So there is an interplay between the components' statuses and the dynamics of the physical variables. In order to accurately model the evolution of the physical variables, we need to model the joint evolution of the physical variables and of the statuses of the components. The vector gathering these two variables is called the state of the system. We denote the state of the system at a time t by Z_t . Denoting X_t the physical variables at a time t , and M_t the statuses of the components at a time t , we have:

$$Z_t = (X_t, M_t). \tag{1.79}$$

We consider the system has d real valued physical variables: $X_t \in \mathbb{R}^d$. If the system includes N_c components then the mode takes the following form: $M_t = (M_t^1, M_t^2, \dots, M_t^{N_c})$, where M_t^i is the status of the i -th component. For ease of presentation we consider that the status of a component can be either ON for an activated component, OFF for a deactivated one, or F for a failed component, so that $\mathbb{M} = \{ON, OFF, F\}^{N_c}$. But as long as \mathbb{M} stays countable, it is possible to consider more options for the statuses of the components. For instance, one could consider different regimes of activity instead of the simple status ON , or different types of failure instead of the status F . Note that we can also deal with continuous degradations, like the size of a breach in a pipe for instance: the presence of the degradation can be included in the mode and its size in the position.

1.3.2 A model for reliability assessment

For a given system, we want to assess its reliability r , which is defined by the probability that the system does not fail before a final observation time t_f . Assuming p_D is the probability that the system fails before the final observation time t_f , we have

$$r = 1 - p_D. \quad (1.80)$$

So to assess r is equivalent to assess p_D . That is why in the rest of the thesis we focus on the assessment of p_D the probability of the system failure.

We consider that the state of the system is initiated in a safe state $z_o = (x_o, m_o) \in E$. We want to assess the probability of the system failure which corresponds to the physical variable entering a critical region. We denote by D_X this critical region within \mathbb{R}^d , and by $D = \{(x, m) \in E | x \in D_X\}$ the associated critical region within the state space E . We consider D_X and D are closed subsets of \mathbb{R}^d and E respectively. We define τ_D the reaching time of D by

$$\tau_D = \inf\{t > 0, Z_t \in D\}, \quad (1.81)$$

with the convention that $\tau_D = +\infty$ if $\{t > 0, Z_t \in D\} = \emptyset$. The probability of system failure p_D , is then defined by:

$$p_D = \mathbb{P}(\tau_D < t_f | Z_0 = z_o) = \mathbb{P}_{z_o}(\tau_D < t_f). \quad (1.82)$$

Or denoting \mathcal{D} the set of trajectories which enters D before the time t_f , one can define p_D by:

$$p_D = \mathbb{P}(\mathbf{Z} \in \mathcal{D} | Z_0 = z_o) = \mathbb{P}_{z_o}(\mathbf{Z} \in \mathcal{D}). \quad (1.83)$$

1.3.3 The model

In this section, we present the characteristics of the PDMP associated to our system. Indeed boundaries and state spaces, jumps kernels on boundaries, jump rates, and jump kernels outside boundaries directly come from the system architecture.

Laws of physics give the flow function Φ

When the components of the system are in some given statuses, the laws of physics within the system determine the evolution of the physical variables of the system through a differential equation, like in equation (1.45). The flow function Φ is derived from the solution of each of these equations like in equation (1.47).

Automatic control mechanisms shape the boundaries of state space E

Generally, there are some components in the system which are programmed to activate or deactivate when the position crosses some thresholds. This is what we call an automatic control mechanism. For instance, it is typically what happens with a safety valve on a pipe: when the pressure inside the pipe rises above a safety limit, the valve opens. In other words: when the flow Φ brings the physical values to a safety limit, a component is programmed to change its status. So for each control mechanism, the change in the value of M_t implies a jump in the process $Z_t = (X_t, M_t)$. Therefore, to model such forced jumps we shape the boundaries of the state space so that they coincide with the thresholds triggering the control mechanisms.

Remember that in a PDMP with boundaries the state space is the union of the sets E_m corresponding to the possible states for each mode $m \in \mathbb{M}$. We have:

$$E = \bigcup_{m \in \mathbb{M}} E_m \quad \text{where} \quad \forall m \in \mathbb{M}, E_m = \left\{ (x, m), x \in \Omega_m \right\}. \quad (1.84)$$

Ω_m being an open set included in \mathbb{R}^d . So for each mode $m \in \mathbb{M}$ we specify the set Ω_m such that its boundary $\partial\Omega_m$ matches with the thresholds triggering the control mechanisms. Then the jump corresponding to the control mechanism is triggered when Z_t^- hits ∂E_m . The law of the arrival of the jumps is then given by the jump kernel $\mathcal{K}_{Z_t^-}$.

Control mechanisms and failure-on-demand define the Kernel on boundaries

During a control mechanism triggered at a state z^- , the system is programmed to switch to a safer state $a \in E$. If the only possible output of the control mechanism is the state a , such programming can simply be modeled by setting $\mathcal{K}_{z^-}(\{a\}) = 1$.

But sometimes, in industrial systems, some components have a small probability to fail right at moment of their activation: this phenomenon is called a failure on demand.

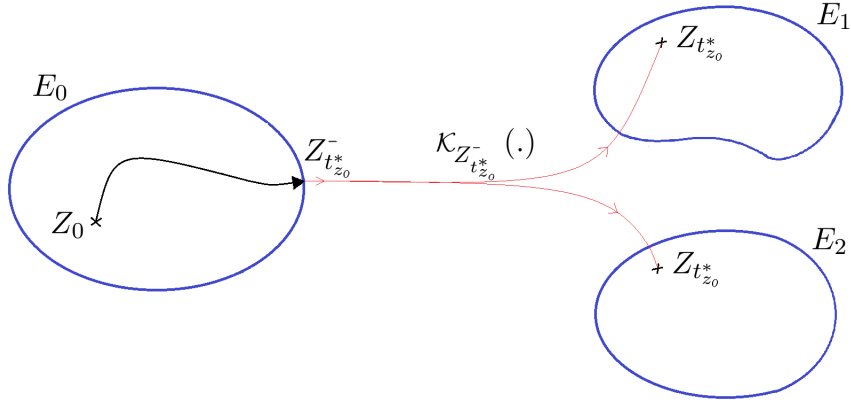


Figure 1.15 – Forced jump at the boundary

Let $A(z^-)$ be the set of the indices of the components required to turn on during the control mechanism. Let $\gamma_{z^-}^i$ be the probability that the i -th component of the system fails during its activation from the state z^- . Let $b \in E$ be a state accessible from z^- in one jump. Let $Fod(z^-, b)$ be the set of indices of the components that fail on demand during a jump from z^- to b . Then we can take:

$$\mathcal{K}_{z^-}(\{b\}) = \prod_{i \in A(z^-) \setminus Fod(z^-, b)} (1 - \gamma_{z^-}^i) \prod_{i \in Fod(z^-, b)} \gamma_{z^-}^i \quad (1.85)$$

to model the control mechanism and the failures-on-demand, from the state z^- . Note for instance, the probability of reaching the safer state a would then be

$$\mathcal{K}_{z^-}(\{a\}) = \prod_{i \in A(z^-)} (1 - \gamma_{z^-}^i). \quad (1.86)$$

The form of the jump Kernel outside of boundaries is determined by the possible transitions

By opposition to jumps at boundaries, the spontaneous jumps are used to model failures and repairs and passive backup mechanisms that do not necessarily happen during a control mechanism. Remember the mode is the vector $M = (M^1, M^2, \dots, M^{N_c}) \in \{\text{ON}, \text{OFF}, \text{F}\}^{N_c}$ gathering the statuses ("ON", "OFF" or "F" for failed) of the N_c components. So a repair, a spontaneous failure, or spontaneous simultaneous failures, are events that can be considered as transitions from a mode m to another mode m^+ . For instance, if the transition corresponds to the repair of the i -th component, the coordinates of m^+ would typically be all identical to those of m , except for the i -th one where we would have $m^i = \text{F}$, as the i -th component is initially failed, and $m^{+i} = \text{OFF}$ or $m^{+i} = \text{ON}$ as the components is then repaired. Whether $m^{+i} = \text{OFF}$ or $m^{+i} = \text{ON}$ will depend on whether the component is turned on or off after its repair. Conversely, for a failure, the coordinate associated to the failing component would switch from the status ON or

OFF to the status F. Simultaneous failures can be modeled by switching the statuses of several components to the failed status F. Backup mechanisms can also be modeled in the transition from m to m^+ , for instance, if the j -th component is programmed to turn on when the i -th component fails m^+ would also differ in its j -th coordinate as we would have $m^j = \text{OFF}$ and $m^{+j} = \text{ON}$.

Although in a PDMP the jumps model a transition from state to state (z to z^+), we often only consider transitions from mode to mode (m to m^+). The reason is that the evolution of the physical variables during a jump is often deterministic once we know the arrival statuses of the components. In most cases, as soon as the mode m^+ is known, we know the only option for the physical variables is to arrive in a point denoted $x_{z \rightarrow m^+}$. If all transitions are of this kind, the jump Kernel then verifies

$$\forall B \in \mathcal{B}(E), \quad \mathcal{K}_z(B) = \sum_{m^+ \in \mathbb{M}} \mathbb{P}(Z_T = z^+ | Z_T^- = z) \delta_{(x_{z \rightarrow m^+}, m^+)}(B), \quad (1.87)$$

where $\delta_{(x_{z \rightarrow m^+}, m^+)}$ is the dirac measure in state $(x_{z \rightarrow m^+}, m^+)$. In this case the jump Kernel is a discrete law of probability.

Though in most cases the kernel is discrete, one can also imagine some cases where the kernel include a continuous part. For instance consider that the physical variables have two dimensions, the first corresponding to pressure on a concrete structure, and the second to the size of a crack in the structure. One can consider that the crack length increases in a jerky way, and that the amplitude of the increase is random and has a continuous law. For a jump triggered from a state $z^- = ((x_1^-, x_2^-), m^-) \in E$ we could have:

$$\forall B \in \mathcal{B}(E), \quad \nu_{z^-}(B) = \int_{\{y_2 > 0\} | ((x_1^-, y_2), m^-) \in B} \mu_{Leb}(dy_2) \quad (1.88)$$

where $\mu_{Leb}(\cdot)$ corresponds to the Lebesgue measure, and

$$\begin{aligned} \forall B \in \mathcal{B}(E), \\ \mathcal{K}_{z^-}(B) = \int_B K_{z^-}(z) \nu_{z^-}(dz) = \int_{\{y_2 > 0\} | ((x_1^-, y_2), m^-) \in B} K_{z^-}(((x_1^-, y_2), m^-)) \mu_{Leb}(dy_2). \end{aligned} \quad (1.89)$$

We think that the cases of non-discrete jump kernels should be rather rare in the reliability analysis field, but PDMPs are also used in other fields, like finance, where non-discrete jump kernels could be more common and for which the use of importance sampling can be of interest too [45]. That is why we keep the most general form of PDMP, which can handle any type of jump kernel.

A link between jump rate and the hazard rates of the possible spontaneous events

Let the system be in the state $z = (x, m)$ at time t , it follows the flow Φ_z until time of the next jump at $t + T$. For a starting state $Z_t = z$, we consider there are a finite number of possible transitions. We denote by J_z the set that indices these possible transitions. For a transition j we define T^j as the time at which this transition occurs. We denote by $A_z^j(u)$ the set of the possible arrivals of the jump when it is triggered by the transition j with $T^j = u$. In order to identify each type of transition we consider that for $i \neq j$ we have $A_z^i(u) \cap A_z^j(u) = \emptyset$. For each transition

$$T^j = \inf\{u \in [0, t_z^*), Z_{t+u}^- = \Phi_z(u), Z_{t+u} \in A_z^i(u)\} \quad (1.90)$$

with the convention that $T^j = +\infty$ if $\{u > t, Z_{t+u}^- = \Phi_z(u), Z_{t+u} \in A_z^i(u)\} = \emptyset$. In order to model the law of T^j we use a state-related intensity function $\lambda^j : E \rightarrow \mathbb{R}^+$ so that:

$$\mathbb{P}(T^j \leq h \mid Z_t = z) = 1 - \exp\left[-\int_0^h \lambda^j(\Phi_z(u)) du\right], \quad (1.91)$$

and we can introduce a time-related intensity function $\lambda_z^j = \lambda^j \circ \Phi_z$ so that:

$$\mathbb{P}(T^j \leq h \mid Z_t = z) = 1 - \exp\left[-\int_0^h \lambda_z^j(u) du\right]. \quad (1.92)$$

The function λ^j would be a failure rate if the transition j from $Z_{t+T^j}^-$ to $Z_{t+T^j} \in A_z^j(T^j)$ is associated to failure, similarly it would be a repair rate if the transition j from $Z_{t+T^j}^-$ to $Z_{t+T^j} \in A_z^j(T^j)$ corresponds to a repair. Note that the time until the next jump, if we start from z , satisfies:

$$T = \min\left[\{T^j, j \in J_z\} \cup \{t_z^*\}\right]. \quad (1.93)$$

In order to compute the cdf of T we make the assumption that the times T^j for $j \in J_z$ are independent knowing $Z_t = z$. This assumption is true if the position gathers all the variables affecting a transition rate. It allows to get the cdf of T as it allows to go from equation (1.94) to (1.95) in the following reasoning:

$$\mathbb{P}(T > h \mid Z_t = z) = \begin{cases} \mathbb{P}(T^j > h, \forall j \mid Z_t = z) & \text{if } h < t_z^*, \\ 0 & \text{if } h \geq t_z^*, \end{cases} \quad (1.94)$$

$$\Leftrightarrow \mathbb{P}(T > h \mid Z_t = z) = \begin{cases} \prod_{j \in J_z} \mathbb{P}(T^j > h \mid Z_t = z) & \text{if } h < t_z^*, \\ 0 & \text{if } h \geq t_z^*, \end{cases} \quad (1.95)$$

$$\Leftrightarrow \mathbb{P}(T \leq h \mid Z_t = z) = \begin{cases} 1 - \prod_{j \in J_z} \mathbb{P}(T^j > h \mid Z_t = z) & \text{if } h < t_z^*, \\ 1 & \text{if } h \geq t_z^*, \end{cases} \quad (1.96)$$

$$\Leftrightarrow \mathbb{P}(T \leq h \mid Z_t = z) = \begin{cases} 1 - \exp \left[- \int_0^h \sum_{j \in J_z} \lambda_z^j(u) du \right] & \text{if } h < t_z^*, \\ 1 & \text{if } h \geq t_z^*. \end{cases} \quad (1.97)$$

So the cdf of T can be expressed with an intensity function λ_z that is the sum of the intensities of all the possible transitions:

$$\mathbb{P}(T \leq h \mid Z_t = z) = \begin{cases} 1 - \exp \left[- \int_0^h \lambda_z(u) du \right] & \text{if } h < t_z^*, \\ 1 & \text{if } h \geq t_z^*, \end{cases}, \quad (1.98)$$

where

$$\forall u \in [0, t_z^*), \quad \lambda_z(u) = \sum_{j \in J_z} \lambda_z^j(u). \quad (1.99)$$

Similarly the state related intensity satisfies

$$\forall z \in E, \quad \lambda(z) = \sum_{j \in J_z} \lambda^j(z). \quad (1.100)$$

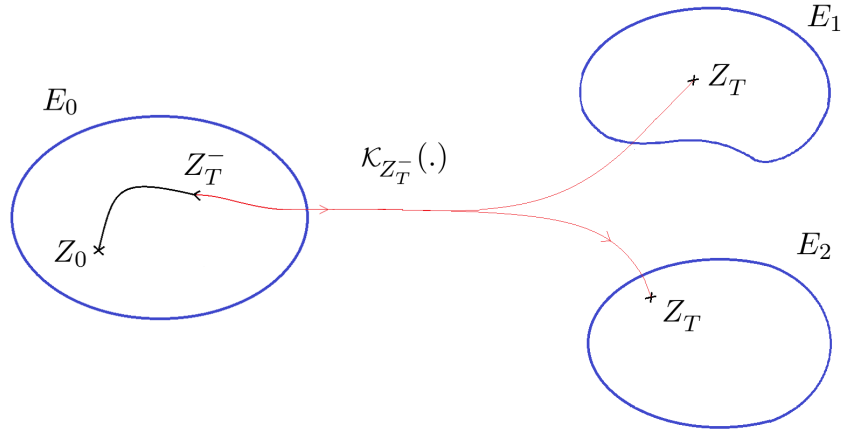


Figure 1.16 – A spontaneous jump before the boundary

Probability of a transition knowing T

Note that equations (1.98) and (1.99) characterize the time of the next jump, but do not tell which transition is being triggered. Knowing the timing of the jump, the nature of the jump is still random, and the index of the transition triggered is then given by the transition Kernel $\mathcal{K}_{Z_{t+T}^-}$. One can define this kernel $\mathcal{K}_{Z_{t+T}^-}$ by using the assumption that

the times T^j are independent knowing $Z_t = z$. As $T = \min [\{T^j, j \in J_z\} \cup \{t_z^*\}]$, we have:

$$\mathbb{P}(T = T^i | Z_t = z) = \mathbb{P}(T^i < T^j, \forall j \in J_z, j \neq i | Z_t = z) \quad (1.101)$$

$$= \int_0^{t_z^*} \mathbb{P}(u < T^j, \forall j \in J_z, j \neq i | T^i = u, Z_t = z) f_{T^i|Z_t=z}(u) du \quad (1.102)$$

$$= \int_0^{t_z^*} \left(\prod_{j \in J_z, j \neq i} \mathbb{P}(u < T^j | T^i = u, Z_t = z) \right) f_{T^i|Z_t=z}(u) du \quad (1.103)$$

$$= \int_0^{t_z^*} \left(\prod_{j \in J_z, j \neq i} \exp[-\Lambda_z^j(u)] \right) \lambda_z^i(u) \exp[-\Lambda_z^i(u)] du \quad (1.104)$$

$$= \int_0^{t_z^*} \frac{\lambda_z^i(u)}{\lambda_z(u)} \lambda_z(u) \exp[-\Lambda_z(u)] du \quad (1.105)$$

$$\mathbb{P}(T = T^i | Z_t = z) = \int_0^{t_z^*} \frac{\lambda_z^i(u)}{\lambda_z(u)} \lambda_z(u) \exp[-\Lambda_z(u)] du + 0 \cdot \exp[-\Lambda_z(t_z^*)] \quad (1.106)$$

$$\mathbb{P}(T = T^i | Z_t = z) = \int_0^{t_z^*} \frac{\lambda_z^i(u)}{\lambda_z(u)} f_T(u) du + 0 \cdot f_T(t_z^*), \quad (1.107)$$

where the assumption of independence of the T^j 's knowing $Z_t = z$, implies the equality between (1.102) and (1.103). Therefore we can identify that:

$$\mathbb{P}(T = T^i | T = u, Z_t = z) = \frac{\lambda_z^i(u)}{\lambda_z(u)} \mathbb{1}_{u < t_z^*}. \quad (1.108)$$

Which implies that we have for any departure state $z_{t+u}^- = \Phi_z(u)$, such that $u \in [0, t_z^*)$ the jump kernel satisfies:

$$\forall B \in \mathcal{B}(E), \quad \mathcal{K}_{z_{t+u}^-} (A_z^i(u)) = \frac{\lambda_z^i(u)}{\sum_{j \in J_z} \lambda_z^j(u)}. \quad (1.109)$$

1.3.4 Example of the Heated room system

As an example of a system, we consider a room heated by three identical heaters. X_t represents the temperature of the room at time t . x_e is the exterior temperature. β_1 is the rate of the heat transition with the exterior. β_2 is the heating power of each heater. The differential equation giving the evolution of the position (i.e. the temperature of the room) has the following form:

$$\frac{dX_t}{dt} = \beta_1(x_e - X_t) + \beta_2 \mathbb{1}_{M_i^1 \text{ or } M_i^2 \text{ or } M_i^3 = ON}. \quad (1.110)$$

The heaters are programmed to maintain the temperature within an interval (x_{min}, x_{max})

trajectory with no failure when it is initiated in a state with no degraded component.

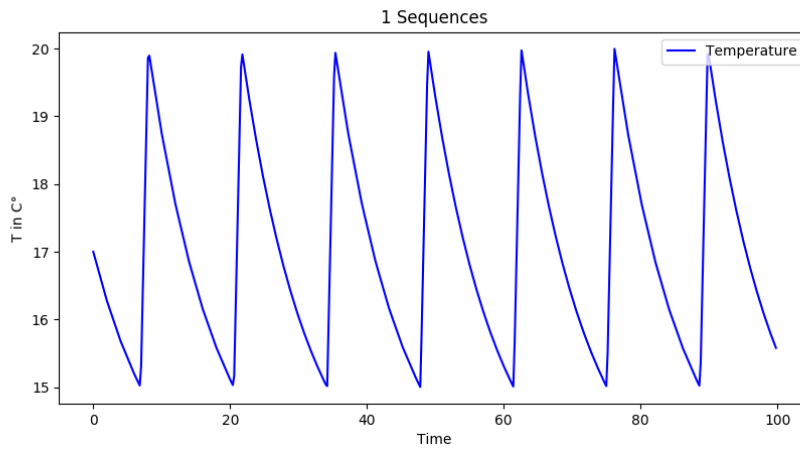


Figure 1.18 – example of evolution of the temperature in the Heated-Room system obtained with PyCATSHOO

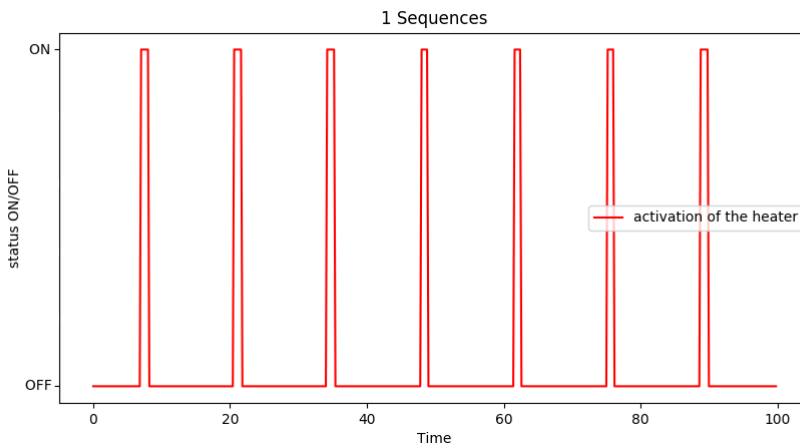


Figure 1.19 – example of evolution of the status of the Heater in the Heated-Room system obtained with PyCATSHOO

1.3.5 Example of the spent fuel pool

The spent fuel pool system is used in a nuclear power plant to stock the spent fuel which is still heating and radioactive. As this system involves many components it would be quite long to expose all the features of its PDMP model. Therefore we just present the structure of the system and the differential equations ruling its main physical variables. The idea here is simply to give the reader a glimpse of the complexity of an industrial system. We refer the reader to [13] for a more complete presentation of this system.

The fuel is placed in the water, because it contains the remaining radiations produced by the fuel. It is important to control the level of the water in the pool so that the fuel stays immersed, without which radiations would be released in the environment. Also it

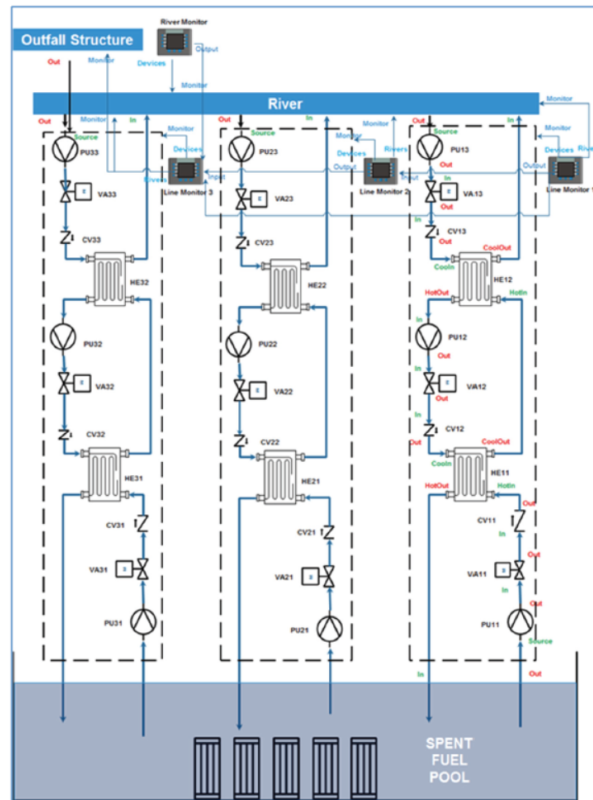


Figure 1.20 – The spent fuel pool system

is important to control the temperature of the pool so that the water does not vaporize.

To do so a cooling system is attached to the pool. This system is designed to cool down the fuel without rejecting any radioactive matter in the river. As the water of the pool is in contact with radioactive matter it can not be released in the nearby river, and it stays in a circuit of water called the primary circuit. This water is heated to the contact of the fuel, and, further in the circuit, it is cooled down by a secondary circuit of water through a heat-exchanger. The heat gathered by the secondary circuit is then absorbed by a third circuit through an other heat-exchanger. The water of the third circuit is taken directly from the nearby river, and goes back to the river once heated. Each circuit has a pump, a valve, and a clap, if one of these elements breaks the water cannot circulate in the circuit. The combination of these three circuits is called a line.

In case the line would fail, it is backed up by two other lines in passive redundancy. All the lines are connected to the river by a pumping system. In case this pumping system fails the third line has an additional backup pumping system. The lines need current to function, and in case of a power grid loss, each line disposes of a switch system which launches a diesel generator that powers the line. The system is schematized in figure 1.20 without its electric-powering part. When the three lines stay failed, the temperature rises to 100 °C, once this evaporating temperature is reached, the level of the water starts to

decrease².

The logic diagram in figure 1.21 represent the logic in the systems. In this diagram if there is an uninterrupted path between the bottom line and the top line it means the main function of the system is fulfilled and that the temperature does not rise nor the level decreases. In this diagram each box represents a component, and acts like a switch that breaks the path when the component is failed. The arrows represent conditional activations, meaning the component pointed by an arrow is activated only if the component at the beginning of the arrow is failed. If a pointed component is not activated it breaks the logic path it is a part of. One can associate a logic function to the figure 1.21 which

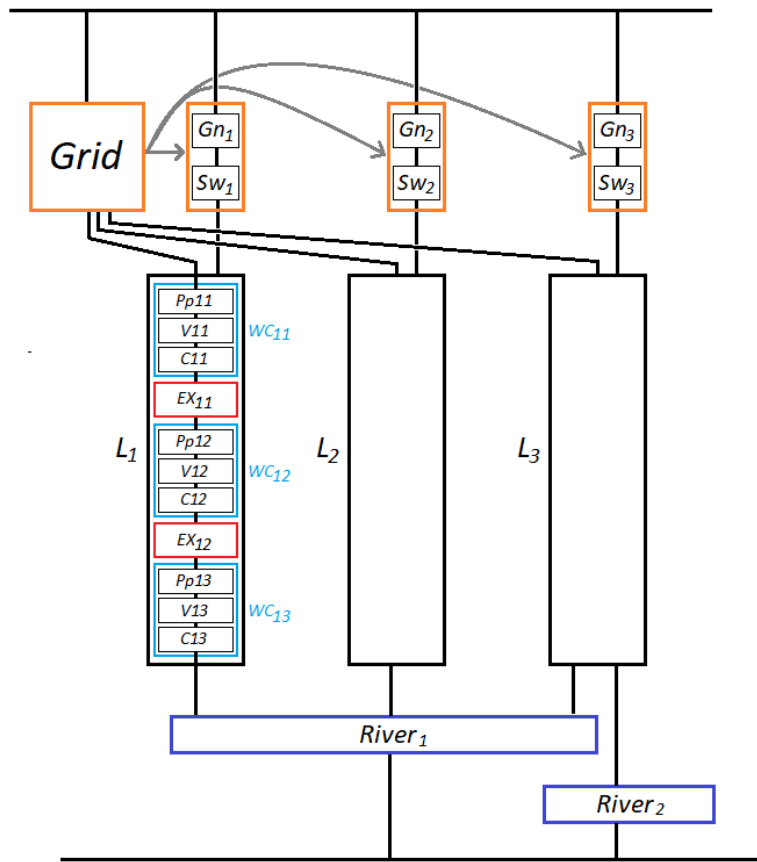


Figure 1.21 – reliability diagram of the static approximation of the spent-fuel-pool system

is true if the function of the system is fulfilled giving the statuses of the components. We denote $L_{pool} : m \in \mathbb{M} \rightarrow \{False, True\}$ the logic function. For each component we define Boolean indicators that are true if they are failed and false otherwise: $Grid$ is the Boolean indicating that the power is no longer available from the grid, Sw_i indicates if the switch of the i -th line is failed, Gn_i indicates if the diesel generator of the i -th line is failed, Lni indicates if the i -th line is failed, Ex_{ij} indicates if the j -th exchanger of the i -th line is failed, Rvi indicates the i -th pump on the river is failed, Wc_{ij} indicates if the

2. In a real spent-fuel-pool at this stage water would be added to the pool to compensate the loss of level, but we consider in this thesis a spent-fuel-pool system without this function of level management.

j -th water circuit of the i -th line is failed, Pp_{ij} indicates if the pump of the j -th water circuit of the i -th line is failed, Va_{ij} indicates if the valves of the j -th water circuit of the i -th line is failed, Cl_{ij} indicates if the clap of the j -th water circuit of the i -th line is failed. Considering multiplication of Booleans as an AND operator and the sum as an OR operator, we have:

$$\begin{aligned} L_{pool}(M) = & Grid [Rv_1(Ln_3 + Sw_3 + Gn_3 + Rv_2) \\ & + (Ln_1 + Sw_1 + Gn_1)(Ln_2 + Sw_2 + Gn_2)(Ln_3 + Sw_3 + Gn_3)] \\ & + Rv_1(Ln_3 + Rv_2), \\ & + (Ln_1)(Ln_2)(Ln_3), \end{aligned} \quad (1.111)$$

$$\text{where } Ln_i = Wc_{i1} + Ex_{i1} + Wc_{i2} + Ex_{i2} + Wc_{i3} \quad (1.112)$$

$$\text{and } Wc_{ij} = Pp_{ij} + Va_{ij} + Cl_{ij} \quad (1.113)$$

We denote by X_t^1 the temperature of the pool at time t in Celsius degrees, and X_t^2 the water level of the pool. P is the residual power of the fuel in the pool, C is the mass heat capacity, ρ is the density of the water, S is the area of the surface of water in contact with the air, Q_{in} is the entering debit water, T_{in} is the temperature of the entering water, respectively Q_{out} is the debit of water leaving the pull, and T_{out} is the temperature of the water leaving the pool, l is the mass latent heat of vaporization. When the temperature is below 100 °C or when $P + \rho C(Q_{in}T_{in} - Q_{out}T_{out}) < 0$, the differential system ruling the evolution of $X_t = (X_t^1, X_t^2)$ is:

$$\forall X_t^1 \in (0, 100), \text{ and } X_t^2 > 0, \quad \frac{dX_t^1}{dt} = \frac{P + \rho C(Q_{in}T_{in} - Q_{out}T_{out})}{\rho C S X_t^2} \quad (1.114)$$

$$\frac{dX_t^2}{dt} = 0. \quad (1.115)$$

When the temperature of the water is at 100 °C and when $P + \rho C(Q_{in}T_{in} - Q_{out}T_{out}) > 0$ the differential system is

$$\text{for } X_t^1 = 100, \text{ and } X_t^2 > 0, \quad \frac{dX_t^1}{dt} = 0 \quad (1.116)$$

$$\frac{dX_t^2}{dt} = -\frac{P + \rho C(Q_{in}T_{in} - Q_{out}T_{out})}{\rho C S l}. \quad (1.117)$$

Monte-Carlo methods for rare events

As the previous Chapter presented our model for the simulation output, the present chapter deals with the simulation methods. We start this chapter by introducing the naive Monte-Carlo method and its rare event issue, and then we present the importance sampling method and the interacting particle system method. Our goal here is not to review all the methods used in reliability analysis, but rather to introduce the methods that we adapt to PDMP in parts II and IV. We refer the readers to [3, 47, 8, 46, 32] for reviews of methods used in rare event analysis.

2.1 The Monte-Carlo method and its rare event issue

Let Y be a random variable with values in a measurable space (A, \mathcal{A}) , let ζ be a reference measure on (A, \mathcal{A}) , let f be the density of Y with respect to ζ , and let h be a measurable function from A to \mathbb{R} .

The Monte-Carlo method allows the estimation of an expectation defined by:

$$p = \mathbb{E}_f[h(Y)] = \int_A h(y)f(y)d\zeta(y). \quad (2.1)$$

The estimator of Monte-Carlo takes the following form:

$$\hat{p}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} h(Y_i), \quad \text{where } Y_i \stackrel{iid}{\sim} f. \quad (2.2)$$

Theorem 3. *If $\mathbb{E}[|h(X)|] < \infty$ the strong law of large numbers implies that the Monte-Carlo estimator is strongly consistent:*

$$\hat{p}_f \xrightarrow[N_f \rightarrow +\infty]{a.s.} p. \quad (2.3)$$

Theorem 4. *If $\mathbb{E}[h^2(Y)] < \infty$, the Monte-Carlo estimator also satisfies a central limit theorem (CLT):*

$$\sqrt{N_f}(\hat{p}_f - p) \xrightarrow[N_f \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_f^2), \quad (2.4)$$

where $\sigma_f^2 = \text{Var}_f(h(Y))$.

This CLT allows to define an asymptotic confidence interval of level $1 - \alpha$ for the

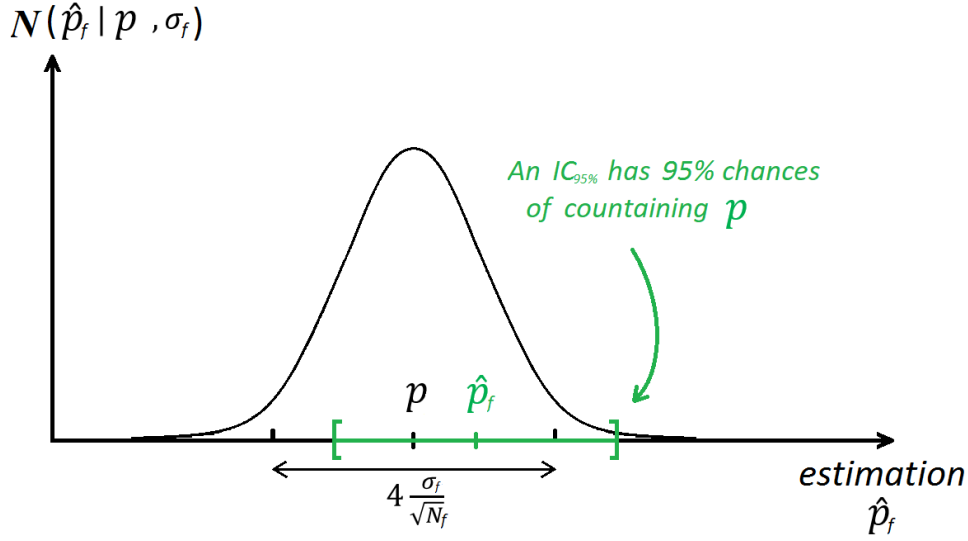


Figure 2.1 – Normally distributed estimator and confidence interval

quantity p which is defined by:

$$IC_{1-\alpha} = \left[\hat{p}_f - q_{\frac{1-\alpha}{2}} \sqrt{\frac{\hat{\sigma}_f^2}{N_f}}, \hat{p}_f + q_{\frac{1-\alpha}{2}} \sqrt{\frac{\hat{\sigma}_f^2}{N_f}} \right], \quad (2.5)$$

where $\hat{\sigma}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} (h(Y_i) - \hat{p}_f)^2$ and $q_{\frac{1-\alpha}{2}}$ is the quantile of level $\frac{1-\alpha}{2}$ of a $\mathcal{N}(0, 1)$ normal distribution. According to the Theorem 2.4 and Slutsky's theorem, $IC_{1-\alpha}$ is an asymptotic confidence interval of level $1 - \alpha$, as

$$\lim_{N_f \rightarrow \infty} \mathbb{P}(p \in IC_{1-\alpha}) = 1 - \alpha. \quad (2.6)$$

Most of the time, one considers a confidence interval of level 95%. In this case $q_{\frac{\alpha}{2}} \simeq 1.96$, and:

$$IC_{95\%} = \left[\hat{p}_f - 1.96 \sqrt{\frac{\hat{\sigma}_f^2}{N_f}}, \hat{p}_f + 1.96 \sqrt{\frac{\hat{\sigma}_f^2}{N_f}} \right]. \quad (2.7)$$

If we want to get a confidence interval of level $1 - \alpha$ of size smaller than $L > 0$, we have to set a number of simulations verifying:

$$N_f \geq \frac{4 q_{\frac{\alpha}{2}}^2 \sigma_f^2}{L^2} \quad (2.8)$$

The more we want to be accurate and have a narrow confidence interval, the higher the number of simulations needs to be. We can even see in the inequality (2.8), that there is a quadratic relation between the number of simulations required and the inverse of width of the CI $\frac{1}{L}$.

2.1.1 Computational burden of the Monte-Carlo estimator

For the Monte-Carlo estimator, when $p_f \ll 1$ the coefficient of variation satisfies

$$CV = \sqrt{\frac{p_f(1-p_f)}{N_f} \frac{1}{p_f}} \simeq \frac{1}{\sqrt{p_f N_f}}. \quad (2.9)$$

Making the approximation $CV = \frac{1}{\sqrt{p_f N_f}}$, a condition $CV < c$ where $c \in \mathbb{R}^+$ implies that

$$N_f > \frac{1}{c^2} \frac{1}{p_f}. \quad (2.10)$$

So when the value of p_f is close to zero, the condition (2.10) shows that the number of simulations needed must be very high. Running that many simulations being computationally intensive, the Monte-Carlo method is therefore ill-suited to rare event analysis. In order to avoid the computational burden of the Monte-Carlo method, one generally prefers using a variance reduction method.

2.1.2 The principle of variance reduction method

The principle of a variance reduction method is to replace the Monte-Carlo estimator by an estimator satisfying a CLT and for which the asymptotic variance is smaller. The interest of taking such an estimator is that it requires less simulation runs to reach a given precision, which eventually accelerates the estimation.

A more accurate estimator means less simulation runs

Denote by \hat{p} an estimator of p based on N simulation runs, and such that

$$\sqrt{N}(\hat{p} - p) \xrightarrow[N \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma^2). \quad (2.11)$$

If we choose both N and N_f to get a length of confidence interval L , we would have:

$$2q_{\frac{\alpha}{2}} \sqrt{\frac{\sigma_f^2}{N_f}} = L = 2q_{\frac{\alpha}{2}} \sqrt{\frac{\sigma^2}{N}}, \quad (2.12)$$

and therefore

$$N = \frac{\sigma^2}{\sigma_f^2} N_f. \quad (2.13)$$

We can see that, for a given accuracy, when $\sigma^2 < \sigma_f^2$ we have $N < N_f$. Equation (2.13) even indicates that reducing the variance by a factor x reduces the required number of simulations by a factor x^2 .

Note that this heuristic reasoning (and especially the equation (2.12)) is valid only if

we assume that the estimators have reached their asymptotic regime. So even if N can be chosen $\frac{\sigma^2}{\sigma_f^2}$ times smaller than N_f , N still has to be big enough so that the Normal approximation on $\sqrt{N}(\hat{p} - p)$ and $\sqrt{N_f}(\hat{p}_f - p)$ are satisfied.

Efficiency and comparison of estimators

In variance reduction methods, getting an estimator with a reduced asymptotic variance is often done by altering the simulation process, and this can sometimes slow down the simulation process. In practice, there is a trade-off between reducing the number of simulation runs and increasing the computational time of each simulation run. For this reason, in [36] the efficiency of an estimator is defined by:

$$eff = \frac{1}{\tau\sigma^2}, \quad (2.14)$$

where τ is the mean computational time for a simulation run. This quantity eff can be interpreted as the contribution of a second of computation to the precision of the estimator. It is a good indicator to compare estimators in practice.

In this thesis we focus on the reduction of the variance, putting aside the mean computational time for a simulation run, mainly because the variance reduction often compensates for the increase of the time of simulation runs, and because reducing the mean computational time for a simulation run is essentially a code optimization issue, which is not our main domain of expertise. This is why we will sometimes compare estimators directly with their variances and not their efficiencies.

2.2 Importance Sampling

In this section we present the method we adapt for PDMP in the Part 2. Namely we present the importance sampling method and the adaptive Cross-Entropy method which we will use to optimize the variance of the importance sampling estimator. For a more complete review of the different uses of importance sampling, and of different ways to optimize it we refer the reader to [51].

2.2.1 Principle

Like the Monte-Carlo method the importance sampling estimate quantities of the form:

$$p = \mathbb{E}_f[h(Y)]. \quad (2.15)$$

With the Monte-Carlo method, when $h = \mathbb{1}_D$ and $p \simeq 10^{-a} < 10^{-4}$, having $Y_i \in D$ is not likely. Consequently, when N_f is not big enough, \hat{p}_f is likely to be null. In practice,

if $p \simeq 10^{-a}$, we often have to simulate approximately 10^a in order to get at least one simulation Y_i in D , and 10^{a+2} simulations are required to get a good estimation of p . To avoid waiting this time before getting a first Y_i in D , the first idea of the Importance Sampling is to simulate from an other random variable \tilde{Y} , for which the region D is more likely. We give more "importance" to a specific region. Let $(\tilde{Y}_i)_{i=1..N}$ be a sequence of N independent random variables and distributed like \tilde{Y} . We denote by g their density with respect to the reference measure ζ . Replacing the Y_i 's by other random variables \tilde{Y}_i 's distributed as \tilde{Y} would yield an estimator $\frac{1}{N} \sum_{i=0}^N \mathbb{1}_D(\tilde{Y}_i)$ that is less likely to be null. This estimator would provide a positive estimation sooner but would unfortunately be biased. The second idea of the Importance Sampling is to add a weight to each simulation output in order to correct this induced bias. If a simulation has been drawn with x times more chances then it is weighted by $\frac{1}{x}$. So a simulation \tilde{Y}_i is weighted by a likelihood ratio $\frac{f(\tilde{Y}_i)}{g(\tilde{Y}_i)}$, which yields the following estimator:

$$\hat{p}_g = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_D(\tilde{Y}_i) \frac{f(\tilde{Y}_i)}{g(\tilde{Y}_i)}, \quad \text{where } \tilde{Y}_i \stackrel{iid}{\sim} g. \quad (2.16)$$

Then these ideas can be generalized for any function h .

2.2.2 The Importance Sampling estimator

The importance sampling estimator of $p = \mathbb{E}_f[h(Y)]$ is

$$\hat{p}_g = \frac{1}{N} \sum_{i=1}^N h(\tilde{Y}_i) \frac{f(\tilde{Y}_i)}{g(\tilde{Y}_i)}, \quad \text{where } \tilde{Y}_i \stackrel{iid}{\sim} g. \quad (2.17)$$

Theorem 5. *This estimator is unbiased, if and only if for ζ -almost every y*

$$g(y) = 0 \Rightarrow h(y)f(y) = 0. \quad (2.18)$$

Proof. Indeed in this case the reasoning below provides the equality (2.19):

$$\begin{aligned} p &= \mathbb{E}_f[h(Y)] \\ &= \int_A h(y)f(y)d\zeta(y) \\ &= \int_A h(y)g(y)\frac{f(y)}{g(y)}d\zeta(y) \\ p &= \mathbb{E}_g\left[h(\tilde{Y})\frac{f(\tilde{Y})}{g(\tilde{Y})}\right], \end{aligned} \quad (2.19)$$

and therefore

$$\mathbb{E}[\hat{p}_g] = \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N h(\tilde{Y}_i) \frac{f(\tilde{Y}_i)}{g(\tilde{Y}_i)} \right] = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_g \left[h(\tilde{Y}) \frac{f(\tilde{Y})}{g(\tilde{Y})} \right] = p. \quad (2.20)$$

□

Theorem 6. *The estimator \hat{p}_g is consistent according to the strong law of large number.*

Theorem 7. *If $\mathbb{E}_g \left[h(\tilde{Y})^2 \left(\frac{f(\tilde{Y})}{g(\tilde{Y})} \right)^2 \right] < +\infty$, the estimator \hat{p}_g satisfies a CLT:*

$$\sqrt{N}(\hat{p}_g - p) \xrightarrow[N \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_g^2), \quad (2.21)$$

$$\text{where } \sigma_g^2 = \mathbb{E}_g \left[h(\tilde{Y})^2 \left(\frac{f(\tilde{Y})}{g(\tilde{Y})} \right)^2 \right] - p^2. \quad (2.22)$$

The variance σ_g can be estimated by the quantity:

$$\hat{\sigma}_g^2 = \frac{1}{N-1} \sum_{i=1}^N \left(h(\tilde{Y}_i) \frac{f(\tilde{Y}_i)}{g(\tilde{Y}_i)} - \hat{p}_g \right)^2 \quad (2.23)$$

Theorem 8. *The estimator $\hat{\sigma}_g^2$ is an unbiased estimator of σ_g^2 , and it is consistent.*

2.2.3 Dynamical importance sampling

Note that the application scope of the importance sampling method is very general as it can be used with any random variables Y and \tilde{Y} as soon as we can define the likelihood ratio $\frac{f(\tilde{Y})}{g(\tilde{Y})}$ and that $\mathbb{E}_g \left[h(Y)^2 \left(\frac{f(Y)}{g(Y)} \right)^2 \right] < +\infty$. In particular it can deal with the dynamic cases: For instance if we consider that $Y = (X_0, X_1, \dots, X_k)$ and $\tilde{Y} = (\tilde{X}_0, \tilde{X}_1, \dots, \tilde{X}_k)$ are a Markov chain with respective kernels densities $Q(X_{j+1}, X_j)$ and $\tilde{Q}(\tilde{X}_{j+1}, \tilde{X}_j)$ with respect to a commune measure, assuming \tilde{Y} is absolutely continuous with respect to Y then the likelihood ratio exists and it is defined by:

$$\frac{f(\tilde{Y})}{g(\tilde{Y})} = \prod_{j=0}^k \frac{Q(\tilde{X}_{j+1}, \tilde{X}_j)}{\tilde{Q}(\tilde{X}_{j+1}, \tilde{X}_j)}. \quad (2.24)$$

Importance sampling can also be used for others sub-classes of PDMPs like continuous time Markov chains, or homogeneous and inhomogeneous Poisson processes [29]. In part II we will show that it can also be generalized when Y is a PDMP trajectory.

2.2.4 Variance and optimal density

The variance of the importance sampling estimator is $\text{Var}(\hat{p}_g) = \frac{\sigma_g^2}{N}$, where the value of σ_g^2 depends on the importance density g . If g is well chosen the variance of the importance sampling estimator can be smaller than the variance of the Monte-Carlo estimator. Indeed:

$$\begin{aligned}\sigma_g^2 &= \int_A h(y)^2 \left(\frac{f(y)}{g(y)} \right)^2 g(y) d\zeta(y) - p^2 \\ &= \int_A h(y)^2 \frac{f(y)}{g(y)} f(y) d\zeta(y) - p^2 \\ &= \mathbb{E}_f \left[h(Y)^2 \frac{f(Y)}{g(Y)} \right] - p^2,\end{aligned}\tag{2.25}$$

and so:

$$\sigma_g^2 - \sigma_f^2 = \mathbb{E}_f \left[h(Y)^2 \left(\frac{f(Y)}{g(Y)} - 1 \right) \right].\tag{2.26}$$

Which gives the following theorem:

Theorem 9. *The variance of the importance sampling estimator is smaller than the one of the Monte-Carlo estimator if and only if*

$$\mathbb{E}_f \left[h(Y)^2 \left(\frac{f(Y)}{g(Y)} - 1 \right) \right] < 0.\tag{2.27}$$

There is an optimal density, that minimizes the variance:

Theorem 10 (optimal density). *The importance density g^* such that*

$$g^*(y) = \frac{|h(y)|f(y)}{\mathbb{E}[|h(Y)|]},\tag{2.28}$$

minimizes the variance as:

$$\sigma_{g^*}^2 = \mathbb{E}_f [|h(Y)|^2] - p^2,\tag{2.29}$$

and for any density g such that $\text{supp}(h.f) \subset \text{supp}(g)$

$$\sigma_{g^*}^2 \leq \sigma_g^2.\tag{2.30}$$

Proof. If g is a density such that $\text{supp}(fh) \subset \text{supp}(g)$ we

$$p^2 + \sigma_{g^*}^2 = \mathbb{E}_{g^*} \left[h(\tilde{Y})^2 \frac{f(\tilde{Y})^2}{g^*(\tilde{Y})^2} \right] \quad (2.31)$$

$$= \int h(y)^2 \frac{f(y)^2}{g^*(y)} d\zeta(y) \quad (2.32)$$

$$= \mathbb{E}_f[|h(Y)|] \int h(y)^2 \frac{f(y)^2}{|h(y)|f(y)} d\zeta(y) \quad (2.33)$$

$$= \mathbb{E}_f[|h(Y)|]^2 \quad (2.34)$$

$$= \mathbb{E}_g \left[|h(\tilde{Y})| \frac{f(\tilde{Y})}{g(\tilde{Y})} \right]^2. \quad (2.35)$$

$$(2.36)$$

So by Cauchy-Schwarz inequality

$$p^2 + \sigma_{g^*}^2 \leq \mathbb{E}_g \left[|h(\tilde{Y})|^2 \frac{f(\tilde{Y})^2}{g(\tilde{Y})^2} \right] \quad (2.37)$$

$$\leq \mathbb{E}_g \left[h(\tilde{Y})^2 \frac{f(\tilde{Y})^2}{g(\tilde{Y})^2} \right] \quad (2.38)$$

$$\leq p^2 + \sigma_g^2 \quad (2.39)$$

$$\text{and so } \sigma_{g^*}^2 \leq \sigma_g^2 \quad (2.40)$$

Which proves theorem 10. □

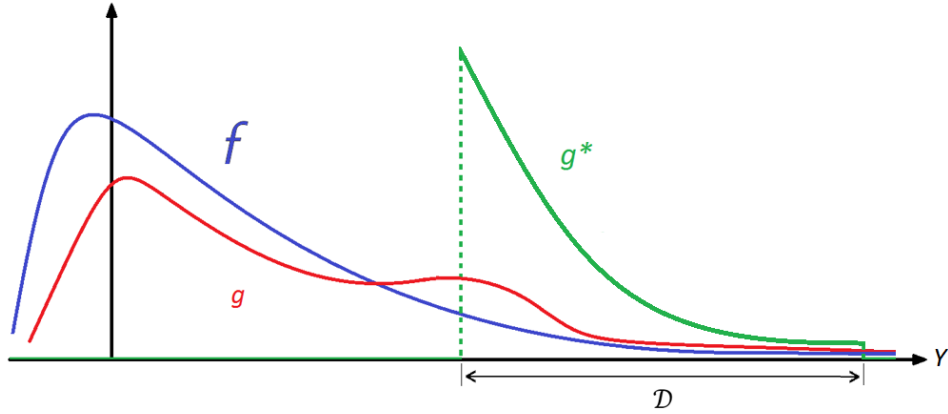
In some case, it is even theoretically possible to bring $\text{Var}(\hat{p}_g)$ to zero as expressed in the following corollary:

Corollary 2 (optimal density for constant sign functions). If $\forall Y \in D, h(y) \geq 0$ or if $\forall y \in A, h(y) \leq 0$ then

$$g^*(y) = \frac{|h(y)|f(y)}{p}, \quad \text{and} \quad \text{Var}(\hat{p}_{g^*}) = 0. \quad (2.41)$$

This corollary applies in particular in the reliability assessment case where $h = \mathbb{1}_D$.

In practice it is impossible to set $g = g^*$ as the expression of g^* involves p , the unknown quantity we are trying to estimate and it would require to build a pseudorandom number generator with the distribution with the density g^* . This expression g^* serves more as a guide to specify efficient importance densities that are close to this optimal density, than as an actual candidate density. The figure 2.2 illustrates a possible importance density g along with the original and optimal density f and g^* in the case $h = \mathbb{1}_D$.


 Figure 2.2 – Some importance densities in the case $h = \mathbb{1}_D$

2.2.5 Important practical concerns

In the case $h = \mathbb{1}_D$, the optimal density g^* indicates that we have to increase the probability of drawing in D , but we have to do it homogeneously, keeping the importance density proportional to f . In the general case, g^* indicates that we have to increase the probability in $\text{supp}(h)$, keeping the importance density proportional to $h \cdot f$. This is well illustrated by following property:

Property 5. If $\forall y \in A, h(y) \geq 0$, and

$$\forall y \in \text{supp}(h), \quad g(y) \geq \frac{h(y)f(y)}{C+p} \quad (2.42)$$

where $C \in [0, 1 - p]$, then

$$\sigma_g^2 \leq Cp, \quad \text{and} \quad \text{Var}(\hat{p}_g) \leq \frac{Cp}{N}. \quad (2.43)$$

The closer the constant C is to 0, the better the variance reduction is. Which means that the more we increase the density g while keeping a proportionality with $h \cdot f$, the better the variance reduction is.

Conversely, in the case $h = \mathbb{1}_D$, increasing the probability inhomogeneously in D ($D = \text{supp}(h)$), can lead to a slow convergence and to **misleading results in practice**. Consider that the regions D_1 and D_2 form a partition of D , and let $p_1 = \mathbb{E}[\mathbb{1}_{D_1}(Y)]$ and $p_2 = \mathbb{E}[\mathbb{1}_{D_2}(Y)]$ so that $p = p_1 + p_2$. Assume the importance density is such that the probability of the region D_2 is increased as much as possible but not the region D_1 :

$$g(y) = \mathbb{1}_{D_1}(y)f(y) + \frac{1-p_1}{p_2}\mathbb{1}_{D_2}(y)f(y). \quad (2.44)$$

In this case, we assume that $p_1 \ll 1$ and $p_2 \ll 1$, and that $p_1 > p_2$. So the asymptotically

variance satisfies

$$\sigma_g^2 = p_1 + \frac{p_2^2}{1 - p_2} - p^2 \simeq p_1. \quad (2.45)$$

If p_1 and p are small and of the same order of magnitude, this variance is comparable to the asymptotic variance of the Monte-Carlo method $p(1 - p)$, so g is not such a good importance density in this case.

An other issue with such an unbalanced importance density is that we may not obtain the right results if we do not take a sufficient number of simulations. The estimator is still convergent but its convergence is too slow. Indeed if $N \ll \frac{1}{p_1}$ we have small chances of drawing a simulation in D_1 . For instance, if only N_2 simulations of the sample are drawn in D_2 and none is drawn in D_1 , then the result is underestimated as then $\hat{p}_g = \frac{N_2}{N} \frac{p_2}{1 - p_1}$ and the empirical variance does not reflect the asymptotic variance at all as: $\hat{\sigma}_g^2 = 0 \neq \sigma_g^2$, hence the slow convergence.

The example we took is deliberately extreme, but the consequence of taking an importance density not proportional to $h.f$ can affect real cases: this typically happens when the region $\text{supp}(h)$ is not connex, or when $h.f$ is multimodal, and when the shape of importance density considered are not flexible enough to match the shape of $h.f$ while increasing the probability mass in $\text{supp}(h)$. If the importance density only favors a mode of $h.f$, or only one region of D , then a too small number of simulations will yield an empirical variance that does not represent the asymptotic variance and this will give an underestimated probability estimate. Consequently the practitioners must try to address this issue of homogeneously increased importance densities when they consider any importance density.

2.3 Variance optimization methods for importance sampling

As we lack mathematical tools to minimize the variance over the set of the possible importance densities, one often restricts the minimization to a set of parametric densities, which allows optimizing over a set of parameters. In the case where $h = \mathbb{1}_D$, and Y involves an empirical mean of random variables the choice of the parametric densities can be based on a large deviation analysis, see [28, 29, 50], though this supposes that the random variable Y is defined in an Euclidean space. In other cases the parametric family is generally built using the practitioner's knowledge about g^* , or, if no valuable knowledge about g^* is available, by taking a flexible family of parametric importance densities.

2.3.1 Variance optimization: the Cross-Entropy method

Once the parametric family of importance densities is chosen, we can use different methods to find the best parameters. We denote by $\alpha \in A_{param} \subseteq \mathbb{R}^n$ the possible parameters, where $n \in \mathbb{N}^*$ and A_{param} is the set of the possible parameters. We denote by $(g_\alpha)_{\alpha \in \mathbb{R}^n}$ the parametric importance densities. In this subsection we present the Cross-Entropy method [20, 34] which optimizes the parameters in order to get the best variance reduction.

The Cross-Entropy method consists firstly in measuring a distance between the densities g_α and g^* , and secondly in minimizing it or in minimizing an approximation of this distance to get the best parameter. The distance between the densities g_α and g^* is measured thanks to the Kullback-Leibler pseudo-metric:

$$D(g^*, g) = \mathbb{E}_{g^*} \left[\log \left(\frac{g^*(Y)}{g(Y)} \right) \right]. \quad (2.46)$$

As $\mathbb{E}_{g^*} [\log (g^*(Y))]$ is a constant (i.e., it does not depend on g), minimizing

$$D(g^*, g_\alpha) = \mathbb{E}_{g^*} [\log (g^*(Y))] - \mathbb{E}_{g^*} [\log (g_\alpha(Y))] \quad (2.47)$$

is equivalent to maximizing

$$\mathbb{E}_{g^*} [\log (g_\alpha(Y))], \quad (2.48)$$

and as $g^*(y) = \frac{h(y)f(y)}{p}$, minimizing $D(g^*, g_\alpha)$ is also equivalent to finding

$$\alpha^* \in \operatorname{argmax}_{\alpha \in A_{param}} \mathbb{E}_f \left[h(Y) \log (g_\alpha(Y)) \right]. \quad (2.49)$$

This can be done if $\mathbb{E}_f [h(Y) \log (g_\alpha(Y))]$ and its derivative with respect to α can be computed analytically.

When $\mathbb{E}_f [h(Y) \log (g_\alpha(Y))]$ can not be computed analytically, one can approximate it empirically by using the following estimator based on $N_{CE,0}$ simulations:

$$\frac{1}{N_{CE,0}} \sum_{i=1}^{N_{CE,0}} h(Y_i) \log (g_\alpha(Y_i)), \quad \text{where } Y_i \sim f. \quad (2.50)$$

A first approximation of α^* , denoted by α_1 , can now be given by solving:

$$\frac{1}{N_{CE,0}} \sum_{i=1}^{N_{CE,0}} h(Y_i) \nabla_\alpha \log (g_\alpha(Y_i)) = 0, \quad \text{where } Y_i \sim f \quad (2.51)$$

Note that, in (2.50), it is important to get at least one realization Y_i such that $h(Y_i) \neq 0$, so that the minimization in (2.51) yields a meaningful minimum α_1 . If the practitioner

chose a number $N_{CE,0}$, there is a chance that all simulations are such that $h(Y_i) = 0$. Therefore in practice, it can be interesting to first chose a number $n_{CE} \in \mathbb{N}^*$ and to gradually increase the number of simulations $N_{CE,0}$ so that n_{CE} of the simulations are such that $h(Y_i) \neq 0$. With a well-chosen parametric family of importance densities this optimization should provide a parameter α_1 for which the importance density g_{α_1} is closer from g^* than f .

Now note that, if we generate the simulations Y_i 's according to the distribution g_{α_1} , so it should provide a better approximation than in equation (2.50), because it is a better importance density than f in this case. By using an importance sampling strategy with the density g_{α_1} , one can improve the approximation of $\mathbb{E}_f[h(Y)\log(g_\alpha(Y))]$ with the following estimator based on $N_{CE,1}$ simulations:

$$\frac{1}{N_{CE,1}} \sum_{i=1}^{N_{CE,1}} h(Y_i) \frac{f(Y_i)}{g_{\alpha_1}(Y_i)} \log(g_{\alpha_1}(Y_i)), \quad \text{where } Y_i \sim g_{\alpha_1}. \quad (2.52)$$

Consequently one can sometimes get a better parameter α_2 by solving:

$$\frac{1}{N_{CE,1}} \sum_{i=1}^{N_{CE,1}} h(Y_i) \frac{f(Y_i)}{g_{\alpha_1}(Y_i)} \nabla_{\alpha} \log(g_{\alpha}(Y_i)) = 0, \quad \text{where } Y_i \sim g_{\alpha_1}. \quad (2.53)$$

This step can even be iterated to get an α_3 , an α_4 , etc... This yields the algorithm 2.1, where we denote f by g_{α_0} . It is also possible to start the algorithm 2.1 with $g_{\alpha_0} \neq f$, in order to start with a better empirical approximation of $\mathbb{E}_f[h(Y)\log(g_\alpha(Y))]$.

Initialization: chose $\alpha_0 \in A_{param}$ and $n_{CE} \in \mathbb{N}^*$ and set $t = 0$, and $\varepsilon > 0$

while $\|\alpha_k - \alpha_{k+1}\| < \varepsilon$ **do**

Set $k = 1$, and generate $Y_1 \sim g_{\alpha_t}$

while $\sum_{i=1}^k \mathbb{1}_{h(Y_i) > 0} < n_{CE}$ **do**

Generate $Y_{k+1} \sim g_{\alpha_t}$

$k := k + 1$

$N = k - 1$

Compute $\alpha_{t+1} = \operatorname{argmin}_{\alpha \in A_{param}} \frac{1}{N} \sum_{i=1}^N h(Y_i) \frac{f(Y_i)}{g_{\alpha_t}(Y_i)} \log(g_{\alpha}(Y_i))$

$t := t + 1$

End: Estimate p with $\hat{p}_{\alpha_{t-1}}$

Table 2.1 – CE algorithm

2.3.2 Important remark on the Cross-Entropy

Note that the theoretical convergence properties of the CE method are not yet fully understood [40] and that the efficiency of this method has only been confirmed by many empirical examples. The convergence on the CE method has only been proven on particular cases, and, in practice, there are cases for which the method does not converge.

This can be due to the fact that $\mathbb{E}_f \left[h(Y) \log(g_\alpha(Y)) \right]$ is poorly approximated because the number of simulations at each step $N_{CE,t}$ is too small, or because the approximation (2.52) is not convex in α and its minimization falls in a local minimum, or because the parametric family of candidate importance densities is ill-suited to the shape of $h(Y)f(Y)$. The approximation of $\mathbb{E}_f \left[h(Y) \log(g_\alpha(Y)) \right]$ is an importance sampling estimation so it suffers from the same practical problems which we mentioned in the subsection 2.2.5. This typically happens when $h(Y)f(Y)$ is multi-modal, which will often be the case in the reliability assessment of a complex system.

It is therefore important NOT to blindly take the result of this method, and to check if its output is robust by running the method several times with different tuning parameters, and to check if the outputs are coherent with our knowledge of the shape of $h(Y)f(Y)$. Though they also do not guarantee the convergence, to go further one could also consider adaptation of the methods that have been proposed to handle multi-modal cases [33, 6, 26, 4].

2.3.3 Adaptive Cross-Entropy

We have seen previously that the CE method can be pursued if we can compute $\operatorname{argmax}_{\alpha \in A_{param}} \mathbb{E}_f \left[h(Y) \log(g_\alpha(Y)) \right]$ analytically, or if we are able to approximate it which requires to generate a simulation such that $h(Y_i) \neq 0$ at least once. For reliability assessment we have $h = \mathbb{1}_D$ and $p = \mathbb{E}_f[h(Y)] = \mathbb{P}(Y \in D)$ is very low, so having $h(Y_i) \neq 0$ (or equivalently $Y_i \in D$) almost never happens. Indeed in practice the number of simulations to realize before getting $Y_i \in D$ is of the order $1/p$, which is often too high compared to the computational capacities. To get the event $h(Y_i) \neq 0$ in a reasonable time is not always possible. In this situation it is preferable to resort to a multi-level importance sampling technique, or an adaptive multi-level importance sampling technique. In this section we present one of these methods which is the adaptive Cross-Entropy method [20].

Assume we dispose of a real function S such that for $s \in \mathbb{R}$:

$$p = \mathbb{P}(Y \in D) = \mathbb{P}(S(Y) \geq s). \quad (2.54)$$

The idea of the multi-level importance sampling is to decompose the problem into a sequence of intermediate problems of reduced complexity. That is to say where the event is obtained faster so that we can get good estimations faster. Let $\tilde{s} \in \mathbb{R}$, so that $\tilde{s} \leq s$. We denote $\tilde{p} = \mathbb{P}(S(Y) \geq \tilde{s})$. The event $S(Y) \geq \tilde{s}$ contains the event $S(Y) \geq s$, so $\tilde{p} \geq p$. Therefore the event $S(Y) \geq \tilde{s}$ is easier to simulate.

The optimal density to estimate \tilde{p} is given by:

$$\tilde{g}^*(y) = \frac{\mathbb{1}_{(S(y) \geq \tilde{s})} f(y)}{\tilde{p}} \quad (2.55)$$

If \tilde{s} is not too far from s the two densities \tilde{g}^* and g^* are very similar (they are proportional on D). Thus when a density g_α gets close to \tilde{g}^* , it also gets close to g^* .

The technique of the multi-level importance sampling consists in taking a sequence of thresholds $(s_t)_{t \geq 0}$ for which, starting from a density $g_{\alpha_{t-1}}$ tuned to estimate $\mathbb{P}(S(Y) \geq s_{t-1})$, one can easily find an efficient parametric density g_{α_t} to estimate $\mathbb{P}(S(Y) \geq s_t)$, and so on.... If the sequence of thresholds tends to s one can construct a sequence of densities g_{α_t} that tends to g_{α^*} (the best density in the parametric family to estimate p).

This method works well when the thresholds $(s_t)_{t \geq 0}$ are chosen so that the probabilities $\mathbb{P}_{\alpha_{t-1}}(S(Y) \geq s_t)$ are not too big, about the order of $\rho = 10^{-2}$ for steps with 10000 simulations, see [20]. Thus, in adaptive multi-level importance sampling s_t is chosen to be equal to the empirical quantile of order $1 - \rho$.

In the end the adaptive multi-level importance sampling consists in applying the algorithm displayed in 2.2, where we denote by $(S_{(1)}, \dots, S_{(N)})$ an ordered sample of the $S(Y_i)$, so that the empirical quantile of order $1 - \rho$ of this sample is $S_{(\lfloor (1-\rho)N \rfloor)}$.

Initialization: choose $\alpha_0 \in A_{param}$ and $\forall t, N_{CE,t} \in \mathbb{N}$ and set $t = 0, s_0 \leq s$
while $s_t \neq s$ **do**
 Generate $(S_i)_{i=1..N_{CE,t}}$ with $S_i = S(Y_i)$ and $Y_i \stackrel{iid}{\sim} g_{\alpha_t}$
 Set $s_t = \min(S_{(\lfloor (1-\rho)N_{CE,t} \rfloor)}, s)$
 Compute $\alpha_{t+1} = \underset{\alpha \in A_{param}}{\operatorname{argmin}} \frac{1}{N_{CE,t}} \sum_{i=1}^{N_{CE,t}} \mathbb{1}_{S(Y_i) > s_t} \frac{f(Y_i)}{g_{\alpha_t}(Y_i)} \log(g_\alpha(Y_i))$
 $t := t + 1$
End: Estimate p with $\hat{p}_{\alpha_{t-1}}$

Table 2.2 – Adaptive CE algorithm

2.4 The interaction particle method

The notations of this section are independent from the notation of the first chapter 1 on PDMPs. Let $(Z_k)_{k \in \{0, \dots, n\}}$ be a Markov chain with values in the measurable spaces (E_k, \mathcal{E}_k) , and with initial law ν_0 , and with a kernel v_k such that for $k > 0$ and for any bounded measurable function $t : E_k \rightarrow \mathbb{R}$

$$\mathbb{E}[t(Z_k)|Z_{k-1}] = \int_{E_k} t(z_k)v_k(dz_k|Z_{k-1}). \quad (2.56)$$

For any bounded measurable function $t : E_0 \times \dots \times E_n \rightarrow \mathbb{R}$ we have:

$$\mathbb{E}[t(Z_0, \dots, Z_n)] = \int_{E_n \times \dots \times E_0} t(z_0, \dots, z_n)v_n(dz_n|z_{n-1}) \cdots v_1(dz_1|z_0)\nu_0(dz_0) \quad (2.57)$$

Let $\mathbf{Z}_k = (Z_0, Z_1, \dots, Z_k)$ be a trajectory of size k , and let $\mathbf{E}_k = E_0 \times E_1 \times \dots \times E_k$ be the set of trajectories of size k that we equip with the product σ -algebra $\mathcal{E}_k = \mathcal{E}_0 \otimes \mathcal{E}_1 \otimes \dots \otimes \mathcal{E}_k$. For $i < j$, consider two trajectories \mathbf{z}_i and \mathbf{z}_j : when it is necessary to differentiate the coordinates of these trajectories we write the coordinates $z_{i,k}$ for $k \leq i$ and $z_{j,k}$ for $k \leq j$ such that $\mathbf{z}_i = (z_{i,0}, z_{i,1}, \dots, z_{i,i})$ and $\mathbf{z}_j = (z_{j,0}, z_{j,1}, \dots, z_{j,j})$. We introduce the Markov Chain of the trajectories $(\mathbf{Z}_k)_{k \geq 0}$ with values in the measurable spaces $(\mathbf{E}_k, \mathcal{E}_k)$, and with the transition kernels \mathbf{V}_k such that:

$$\mathbf{V}_k(d\mathbf{z}_k|\mathbf{z}_{k-1}) = \delta_{\mathbf{z}_{k-1}}(d(z_{k,0}, \dots, z_{k,k-1}))v_k(dz_{k,k}|z_{k-1,k-1}). \quad (2.58)$$

For any bounded measurable function $h : \mathbf{E}_n \rightarrow \mathbb{R}$ we have

$$p = \mathbb{E}[h(\mathbf{Z}_n)] = \iiint_{\mathbf{E}_n \times \dots \times \mathbf{E}_0} h(\mathbf{z}_n) \prod_{k=1}^n \mathbf{V}_k(d\mathbf{z}_k|\mathbf{z}_{k-1})\nu_0(d\mathbf{z}_0). \quad (2.59)$$

The IPS method provides an estimator of p with a different variance than the Monte-Carlo estimator. It was first introduced in [23], and with an alternative formulation in [22].

In the rest of this section 2.4 we use the following notations: We denote by $\mathcal{M}(A)$ the set of bounded measurable functions on a measurable set (A, \mathcal{A}) . If f is a bounded measurable function, and η is a measure we note $\eta(f) = \int f d\eta$. If M is a Markovian kernel, we denote by $M(f)$ the function such that $M(f)(x) = \int f(y)M(dy|x)$, and for a measure η , we denote by ηM the measure such that

$$\eta M(f) = \int \int f(y)M(dy|x)\eta(dx). \quad (2.60)$$

2.4.1 A Feynman-Kac model

The IPS method relies on a Feynman-Kac model [21] whose main characteristics are defined in this subsection. A Feynman-Kac model is a sequence of target distributions and of propagated distributions on the sets $(\mathbf{E}_k, \mathcal{E}_k)$, that are linked by a selection and propagation pattern defined below.

Target and propagated distributions

For each k , $0 \leq k \leq n$, we consider a potential function $G_k : \mathbf{E}_k \rightarrow \mathbb{R}^+$ and we define the *target* probability measure $\tilde{\eta}_k$ on $(\mathbf{E}_k, \mathcal{E}_k)$, such that:

$$\forall B \in \mathcal{E}_k, \quad \tilde{\eta}_k(B) = \frac{\mathbb{E} \left[\mathbb{1}_B(\mathbf{Z}_k) \prod_{s=0}^k G_s(\mathbf{Z}_s) \right]}{\mathbb{E} \left[\prod_{s=0}^k G_s(\mathbf{Z}_s) \right]}. \quad (2.61)$$

For each k , $1 \leq k \leq n$, note that assuming that $\tilde{\eta}_k$ is a probability measure and not the null measure means that $\mathbb{E} \left[\prod_{s=0}^k G_s(\mathbf{Z}_s) \right] > 0$.

For each k , $1 \leq k \leq n$, we define the *propagated* target probability measure η_k on $(\mathbf{E}_k, \mathcal{E}_k)$ such that $\eta_k = \tilde{\eta}_{k-1} M_{k-1}$ and for $k = 0$ we set $\eta_0 = \tilde{\eta}_0$. We have:

$$\forall B \in \mathcal{E}_{k+1}, \quad \eta_{k+1}(B) = \frac{\mathbb{E} \left[\mathbb{1}_B(\mathbf{Z}_{k+1}) \prod_{s=0}^k G_s(\mathbf{Z}_s) \right]}{\mathbb{E} \left[\prod_{s=0}^k G_s(\mathbf{Z}_s) \right]}. \quad (2.62)$$

Link between distributions

Notice that, for $k > 1$, η_k is the propagation of $\tilde{\eta}_{k-1}$ through the transition kernel \mathbf{V}_k as:

$$\eta_k = \tilde{\eta}_{k-1} \mathbf{V}_k. \quad (2.63)$$

Let Ψ_k be the application that transforms a measure η defined on \mathbf{E}_k into a measure $\Psi_k(\eta)$ defined on \mathbf{E}_k and such that

$$\Psi_k(\eta)(f) = \frac{\int G_k(\mathbf{z}) f(\mathbf{z}) d\eta(\mathbf{z})}{\eta(G_k)}. \quad (2.64)$$

We say $\Psi_k(\eta)$ give the selection of η through the potential G_k . Notice that $\tilde{\eta}_k$ is the selection of η_k as

$$\tilde{\eta}_k = \Psi_k(\eta_k). \quad (2.65)$$

The target distributions can therefore be built according to the following pattern of successive selection and propagation:

$$\eta_k \xrightarrow{\Psi_k} \tilde{\eta}_k \xrightarrow{\cdot \mathbf{V}_{k+1}} \eta_{k+1}. \quad (2.66)$$

Unnormalized measures and expectations

We also define the associated unnormalized measures $\tilde{\gamma}_k$ and γ_{k+1} , such that for $f \in \mathcal{M}(\mathbf{E}_k)$:

$$\tilde{\gamma}_k(f) = \mathbb{E} \left[f(\mathbf{Z}_k) \prod_{s=0}^k G_s(\mathbf{Z}_s) \right] \quad \text{and} \quad \tilde{\eta}_k(f) = \frac{\tilde{\gamma}_k(f)}{\tilde{\gamma}_k(\mathbf{1})}, \quad (2.67)$$

and for $f \in \mathcal{M}(\mathbf{E}_{k+1})$:

$$\gamma_{k+1}(f) = \mathbb{E} \left[f(\mathbf{Z}_{k+1}) \prod_{s=0}^k G_s(\mathbf{Z}_s) \right] \quad \text{and} \quad \eta_{k+1}(f) = \frac{\gamma_{k+1}(f)}{\gamma_{k+1}(\mathbf{1})}. \quad (2.68)$$

Remember $p = \mathbb{E}[h(\mathbf{Z}_n)]$. Denoting $f_h(\mathbf{z}_n) = \frac{h(\mathbf{z}_n)}{\prod_{s=0}^{n-1} G_s(\mathbf{z}_s)}$, notice that we have:

$$p = \gamma_n(f_h) = \eta_n(f_h) \prod_{k=0}^{n-1} \eta_k(G_k). \quad (2.69)$$

2.4.2 The IPS algorithm and its estimator

Weighted samples to approximate the distributions and the expectations

The IPS method provides an algorithm to generate weighted samples which approximate the probability measures η_k and $\tilde{\eta}_k$ respectively for each step k , these approximations are then used to provide an estimator of p . For the sample approximating η_k , we denote \mathbf{Z}_k^j the j^{th} trajectory and W_k^j its weight. Respectively in the sample approximating $\tilde{\eta}_k$, we denote $\tilde{\mathbf{Z}}_k^j$ the j^{th} trajectory and \tilde{W}_k^j its associated weight. The trajectories within these samples are sometime called particles. This designation inherits from one of the first case of application of the IPS method that modeled trajectories of particles. For simplicity reasons, in this section, we consider that the samples all contain N trajectories (or particles), but it is possible to modify the sample size at each step, as illustrated in [38]. The empirical approximations of η_k and $\tilde{\eta}_k$ are denoted η_k^N and $\tilde{\eta}_k^N$ and are defined by:

$$\tilde{\eta}_k^N = \sum_{i=1}^N \tilde{W}_k^i \delta_{\tilde{\mathbf{Z}}_k^i} \quad \text{and} \quad \eta_k^N = \sum_{i=1}^N W_k^i \delta_{\mathbf{Z}_k^i}. \quad (2.70)$$

So for all $k \leq n$ and $f \in \mathcal{M}(\mathbf{E}_k)$,

$$\tilde{\eta}_k^N(f) = \sum_{i=1}^N \tilde{W}_k^i f(\tilde{\mathbf{Z}}_k^i) \quad \text{and} \quad \eta_k^N(f) = \sum_{i=1}^N W_k^i f(\mathbf{Z}_k^i). \quad (2.71)$$

By plugging these estimations into equations (2.67) and (2.68), we get estimations for the unnormalized distributions. Noting $\tilde{\gamma}_k^N$ and γ_k^N these estimations, for all $k \leq n$ and

$f \in \mathcal{M}(\mathbf{E}_k)$, we have:

$$\tilde{\gamma}_k^N(f) = \tilde{\eta}_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s) \quad \text{and} \quad \gamma_k^N(f) = \eta_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s). \quad (2.72)$$

In particular if we apply (2.72) to the test function $f_h(\mathbf{Z}_n) = \frac{h(\mathbf{Z}_n)}{\prod_{s=0}^{n-1} G_s(\mathbf{Z}_s)}$, we get an estimator \hat{p}_G of p defined by:

$$\hat{p}_G = \eta_n^N(f_h) \prod_{k=0}^{n-1} \eta_k^N(G_k). \quad (2.73)$$

Sequentially build the samples

The IPS algorithm builds the samples sequentially, alternating between a selection step and a propagation step.

The k^{th} selection step transforms the sample $(\mathbf{Z}_k^j, W_k^j)_{j \leq N}$, into the sample $(\tilde{\mathbf{Z}}_k^j, \tilde{W}_k^j)_{j \leq N}$. This transformation is done with a multinomial resampling scheme. This means that the values of the $\tilde{\mathbf{Z}}_k^j$'s are drawn with replacement from the sample $(\mathbf{Z}_k^j)_{j \leq N}$, each trajectory \mathbf{Z}_k^j having a probability $\frac{W_k^j G_k(\mathbf{Z}_k^j)}{\sum_{i=1}^N W_k^i G_k(\mathbf{Z}_k^i)}$ to be drawn each time. We let \tilde{N}_k^j be the number of times the particle \mathbf{Z}_k^j is replicated in the sample $(\tilde{\mathbf{Z}}_k^j, \tilde{W}_k^j)_j$, so $N = \sum_{j=1}^N \tilde{N}_k^j$. After this resampling the weights \tilde{W}_k^j are set to $\frac{1}{N}$.

The interest of this selection by resampling is that it discards low potential trajectories and replicates high potential trajectories. Thus, the selected sample focuses on trajectories that will have a greater impact on the estimations of the next distributions once extended. If one specifies potential functions that are not positive everywhere, there can be a possibility that at a step k we get $\forall j, G_k(\mathbf{Z}_k^j) = 0$. When this is the case, the probability for resampling can not be defined, the algorithm stops, and we consider that $\forall s \geq k$ the measures $\tilde{\eta}_s^N$ and η_s^N are equal to the null measure.

Then the k^{th} propagation step transforms the sample $(\tilde{\mathbf{Z}}_k^j, \tilde{W}_k^j)_{j \leq N}$, into the sample $(\mathbf{Z}_{k+1}^j, W_{k+1}^j)_{j \leq N}$. Each trajectory \mathbf{Z}_{k+1}^j is obtained by extending the trajectory $\tilde{\mathbf{Z}}_k^j$ one step further using the transition kernel \mathbf{V}_{k+1} . The weights satisfy $W_{k+1}^j = \tilde{W}_k^j$, $\forall j$. Then the procedure is iterated until the step n . The full algorithm to build the samples is displayed in table 2.3. Figure 2.3 displays an example of genealogy of the particles of the samples.

Potential and convergence properties of the estimator

For $k < n$, we denote by $\hat{\mathbf{E}}_k = \{\mathbf{z}_k \in \mathbf{E}_k, G_k(\mathbf{z}_k) > 0\}$ the support of G_k , and we denote $\hat{\mathbf{E}}_n = \{\mathbf{z}_n \in \mathbf{E}_n, h(\mathbf{z}_n) > 0\}$ the support of h . We will make the following assumption on

Initialization: $k = 0$, $\forall j = 1..N$, $\mathbf{Z}_0^j \stackrel{i.i.d.}{\sim} \eta_0$ and $W_0^j = \frac{1}{N}$, and $\tilde{W}_0^j = \frac{G_0(\mathbf{Z}_0^j)}{\sum_s G_0(\mathbf{Z}_0^s)}$
while $k < n$ **do**
 Selection:
 $(\tilde{\mathbf{Z}}_k^j)_{j=1..N} \stackrel{i.i.d.}{\sim} \sum_{i=1}^N \tilde{W}_k^i \delta_{\mathbf{Z}_k^i}$
 $\forall j := 1..N$, $\tilde{W}_k^j := \frac{1}{N}$
 Propagation:
 for $j := 1..N$ **do**
 using the kernel \mathbf{V}_{k+1} , sample the trajectory $\tilde{\mathbf{Z}}_k^j$ to get \mathbf{Z}_{k+1}^j
 set $W_{k+1}^j = \tilde{W}_k^j$ and $\tilde{W}_{k+1}^j = \frac{W_{k+1}^j G_{k+1}(\mathbf{Z}_{k+1}^j)}{\sum_s W_{k+1}^s G_{k+1}(\mathbf{Z}_{k+1}^s)}$
 if $\forall j$, $\tilde{W}_{k+1}^j = 0$ **then**
 $\forall q > k$, set $\eta_q^N = \tilde{\eta}_q^N = 0$ and Stop
 else
 $k := k + 1$

Table 2.3 – IPS algorithm

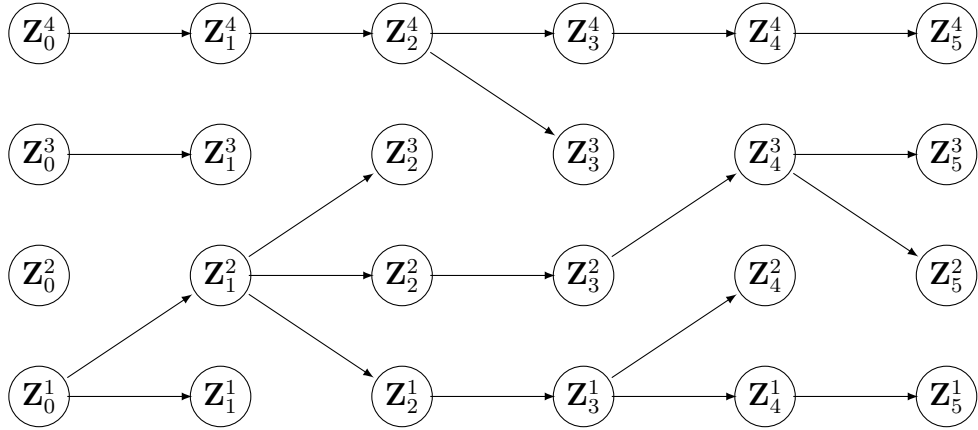


Figure 2.3 – A genealogical tree of an interacting particle system

the potential functions:

$$\exists \varepsilon > 0, \quad \forall k \leq n, \quad \forall \mathbf{z}_{k-1} \in \hat{\mathbf{E}}_{k-1}, \quad \mathbf{V}_k(\hat{\mathbf{E}}_k | \mathbf{z}_{k-1}) > \varepsilon, \quad (\text{G})$$

Theorem 11. *When the potential functions satisfy (G), \hat{p}_G is unbiased and strongly consistent.*

The proof of theorem 11 can be found in [21] chapter 7.

Theorem 12. *When the potential functions are strictly positive:*

$$\sqrt{N}(\hat{p}_G - p) \xrightarrow[N \rightarrow \infty]{d} \mathcal{N}(0, \sigma_{IPS,G}^2) \quad (2.74)$$

where, with the convention that $\prod_{i=0}^{-1} G_i(\mathbf{Z}_i) = \prod_{i=0}^{-1} G_i^{-1}(\mathbf{Z}_i) = 1$:

$$\sigma_{IPS,G}^2 = \sum_{k=0}^n \left\{ \mathbb{E} \left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_i) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n) | \mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] - p^2 \right\}. \quad (2.75)$$

A proof of this CLT can be found in [21] chapter 9.

2.4.3 Estimate the variance of the IPS estimator

The recent work [38] provides two estimators of the variance $\sigma_{IPS,G}^2$. These estimators are weakly convergent in the sense that they converge only in probability to the asymptotic variance.

The study of the variability of the IPS method is connected with the genealogical history of the particles. In order to present the estimator of the variance we need to introduce a few notations to handle the history of the particle. Let A_k^j be the ancestor index of the j^{th} trajectory in the k -th selected sample, such that $\tilde{\mathbf{Z}}_k^j = \mathbf{Z}_k^{A_k^j}$. Let e_n^i be the original ancestor index of i , also called the Eve index of i , it is defined recursively by:

$$e_0^i = i \text{ and for } k, 0 < k \leq n, \quad e_k^i = A_k^{e_{k-1}^i}. \quad (2.76)$$

So e_k^i is the ancestor index at step 0 of the i -th particle of the k -th sample.

The first estimator of the variance $\sigma_{IPS,G}^2$ is:

$$\hat{\sigma}_1^2 = \gamma_n^N (1)^2 \left\{ \eta_n^N (f_h)^2 - \frac{N^{n-1}}{(N-1)^{n+1}} \sum_{i,j:e_n^i \neq e_n^j} f_h(\mathbf{Z}_n^i) f_h(\mathbf{Z}_n^j) \right\}. \quad (2.77)$$

Theorem 13. $\hat{\sigma}_1^2$ is an unbiased and weakly convergent estimator of $\sigma_{IPS,G}^2$.

The second estimator relies on a rewriting of (2.75) in (2.79). For a given vector $b \in \{0, 1\}^n$ we introduce the expectation \mathbb{E}^b with respect to a pair of random vectors $(\mathbf{Y}_k)_{k \leq n}$ and $(\mathbf{Y}'_k)_{k \leq n}$. For a given vector $b \in \{0, 1\}^n$, the law of these vectors are defined as follow: $\mathbf{Y}_0 \sim \eta_0$ and for $k > 0$ $\mathbf{Y}_k \sim \mathbf{V}_k(\cdot | \mathbf{Y}_{k-1})$. If $b_0 = 0$ $\mathbf{Y}'_0 \sim \eta_0$ with \mathbf{Y}'_0 independent of \mathbf{Y}_0 , and if $b_0 = 1$ we take $\mathbf{Y}'_0 = \mathbf{Y}_0$; Similarly for $k > 0$ if $b_k = 0$ take $\mathbf{Y}'_k \sim \mathbf{V}_k(\cdot | \mathbf{Y}'_{k-1})$ with \mathbf{Y}'_k independent of \mathbf{Y}_k , and if $b_k = 1$ we take $\mathbf{Y}'_k = \mathbf{Y}_k$. Now for a function $\tilde{f} : \mathbf{E}_n^2 \rightarrow \mathbb{R}$ we define the quantity $\mu_b(\tilde{f})$ by:

$$\mu_b(\tilde{f}) = \mathbb{E}^b \left[\prod_{k=0}^{n-1} G_k(\mathbf{Y}_k) G_k(\mathbf{Y}'_k) \tilde{f}(\mathbf{Y}_n, \mathbf{Y}'_n) \right], \quad (2.78)$$

For a function $f \in \mathbb{R}^{\mathbf{E}_n}$ we define $f^{\otimes 2}$ as the function of $\mathbb{R}^{\mathbf{E}_n^2}$ such that $f^{\otimes 2}(\mathbf{Y}, \mathbf{Y}') = f(\mathbf{Y})f(\mathbf{Y}')$. Denoting by e_p the vector with the $p+1^{\text{th}}$ coordinate equal to 1 and null

coordinates everywhere else, and denoting by 0_n the null vector of size n , we can rewrite the variance as follows:

$$\sigma_{IPS,G}^2 = \sum_{k=0}^{n-1} \mu_{e_k}(f_h^{\otimes 2}) - \mu_{0_n}(f_h^{\otimes 2}) \quad (2.79)$$

In order to estimate the variance the authors of [38] provide an estimator $\mu_b^N(f^{\otimes 2})$ of the quantity $\mu_b(f^{\otimes 2})$ for any $b \in \{0, 1\}^n$ and define the second estimator by:

$$\hat{\sigma}_2^2 = \sum_{k=0}^{n-1} \mu_{e_k}^N(f_h^{\otimes 2}) - \mu_{0_n}^N(f_h^{\otimes 2}). \quad (2.80)$$

In order to estimate the quantities $\mu_b(\tilde{f})$ they introduce two auxiliary random vectors. These vectors are drawn from the samples built with the IPS algorithm. They are denoted by $\mathbf{Z}^{\mathbf{L}^1} = (\mathbf{Z}_0^{L_0^1}, \dots, \mathbf{Z}_n^{L_n^1})$ and $\mathbf{Z}^{\mathbf{L}^2} = (\mathbf{Z}_0^{L_0^2}, \dots, \mathbf{Z}_n^{L_n^2})$, where $\mathbf{L}^1 = (L_0^1, \dots, L_n^1)$ and $\mathbf{L}^2 = (L_0^2, \dots, L_n^2)$ represent the indices of the trajectories picked in the samples. So these vectors of indices are associated with the genealogical structure of the samples within the algorithm. \mathbf{L}^1 traces backward the ancestral lineage of a trajectory. \mathbf{L}^2 traces backward the ancestral lineage of a trajectory that breaks each time it touches the \mathbf{L}^1 lineage. They are built as follows:

- L_n^1 is drawn uniformly from $\{1, \dots, N\}$, and $\forall p < n$, we set $L_p^1 = A_p^{L_{p+1}^1}$.
- We then build \mathbf{L}^2 knowing \mathbf{L}^1 :
 L_n^2 is drawn uniformly from $\{1, \dots, N\}$, and $\forall p < n$, we set $L_p^2 = A_p^{L_{p+1}^2}$ only when $L_{p+1}^2 \neq L_{p+1}^1$, otherwise in the case $L_{p+1}^2 = L_{p+1}^1$, we draw L_p^2 from $\{1, \dots, N\}$ with a probability

$$\mathbb{P}(L_p^2 = j) = \frac{W_p^j G(\mathbf{Z}_p^j)}{\sum_{s=1}^N W_p^s G(\mathbf{Z}_p^s)}. \quad (2.81)$$

An example of realization of $\mathbf{Z}^{\mathbf{L}^1}$ and $\mathbf{Z}^{\mathbf{L}^2}$ is displayed in figure 2.4.

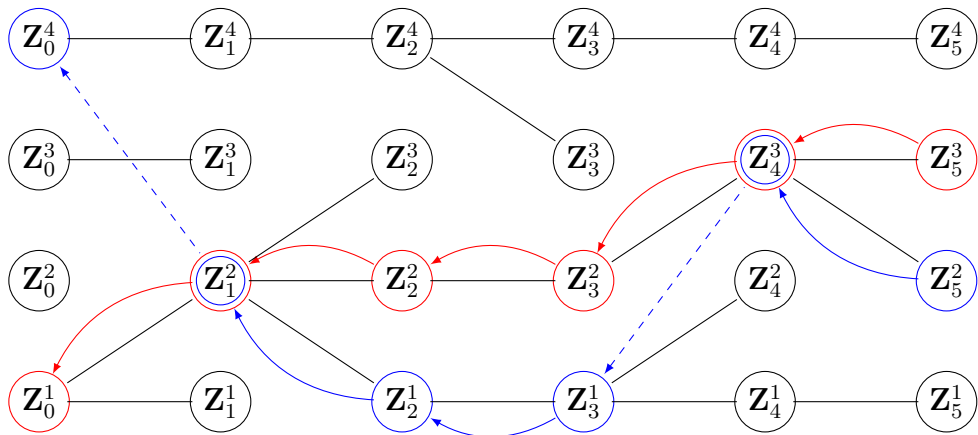


Figure 2.4 – A genealogical tree of an interacting particle system, with a possible lineage $\mathbf{Z}^{\mathbf{L}^1}$ in red and a possible lineage $\mathbf{Z}^{\mathbf{L}^2}$ in blue

We denote by $\underline{A} = \{A_k^j \mid 0 \leq k \leq n, 1 \leq j \leq N\}$ the set of all ancestors, and by $\underline{Z} = \{\mathbf{Z}_k^j \mid 0 \leq k \leq n, 1 \leq j \leq N\}$ the set of all the sampled trajectories. So knowing the structure and leaves of the tree of the particles is equivalent to knowing \underline{A} and \underline{Z} . Denoting by I_b the set of pairs of lineage $(\mathbf{L}^1, \mathbf{L}^2)$ such that $L_p^1 = L_p^2$ if and only if $b_p = 1$, $\mu_b(f)$ can be estimated by:

$$\mu_b^N(f) = \left(\prod_{p=0}^n \frac{N}{(N-1)^{1-b_p}} \right) \gamma_n^N(1)^2 \mathbb{E} \left[\mathbb{1}_{I_b}(L^1, L^2) f(\mathbf{Z}_n^{L^1}, \mathbf{Z}_n^{L^2}) \mid \underline{A}, \underline{Z} \right]. \quad (2.82)$$

An algorithm to compute $\mu_b^N(f)$ can be found in the supplement of [38].

Theorem 14. $\mu_b^N(f)$ is an unbiased and weakly convergent estimator of $\mu_b(f)$.

Therefore we have the following theorem:

Theorem 15. $\hat{\sigma}_2^2$ is an unbiased and weakly convergent estimator of $\sigma_{IPS,G}^2$.

The proof for both theorems 14 and 15 can be found in the supplement of [38].

2.4.4 Classical improvements of the IPS method

The SMC method

We have seen that the resampling steps have the advantage of replicating high potential trajectories and discarding low potential trajectories. However the resampling steps also introduce some additional fluctuations to the estimation (see (2.75)). So we would like to trigger them only when it is judicious. Typically, we do not want to trigger them when the potentials of all the trajectories are similar, as in this case there is not point in discarding or replicating some trajectories over others. In order to avoid pointless resampling, one can trigger the selection step only when the weights are unbalanced. This is done in the SMC algorithm with adaptive resampling [22] presented in table 2.4. In this algorithm, the heterogeneity of the weights is quantified using the effective sample size. At the k^{th} step the effective sample size is defined by:

$$ESS_k = \frac{\left(\sum_{j=1}^N W_k^j G_k(\mathbf{Z}_k^j) \right)^2}{\sum_{i=1}^N \left(W_k^i G_k(\mathbf{Z}_k^i) \right)^2}. \quad (2.83)$$

It is between 1 and N and measures the homogeneity in the candidate weights $\frac{W_k^i G_k(\mathbf{Z}_k^i)}{\sum_j W_k^j G_k(\mathbf{Z}_k^j)}$: when $ESS_k = N$ the weights are perfectly balanced and are all worth $\frac{1}{N}$, and conversely when $ESS_k = 1$ all the weights are null except one, which concentrates the totality of the mass. Therefore, one considers the weights are too unbalanced when $ESS_k < eN$ where $e \in [0, 1]$ is a tuning parameter. Although the resamplings are random in the SMC, in literature people often neglect this point when it comes to variance analysis. In the

```

Initialization:  $k = 0$ ,  $\forall i := 1..N, \mathbf{Z}_0^i = (z_0)$  and  $W_0^i = \frac{1}{N}$ , and  $\tilde{W}_0^i = \frac{G_0(\mathbf{Z}_0^i)}{\sum_j G_0(\mathbf{Z}_0^j)}$ 
while  $k < n$  do
  Selection:
  if  $ESS_k < eN$  then
     $(\tilde{\mathbf{Z}}_k^j)_{j=1..N} \stackrel{i.i.d.}{\sim} \sum_{i=1}^N \tilde{W}_k^i \delta_{\mathbf{Z}_k^i}$ 
    and set  $\forall i = 1..N, \tilde{W}_k^i := \frac{1}{N}$ 
  else
    for  $i := 1..N$  do
      set  $\tilde{\mathbf{Z}}_k^i := \mathbf{Z}_k^i$ 
  Propagation:
  for  $i := 1..N$  do
    sample  $\mathbf{Z}_{k+1}^i \sim \mathbf{V}_k(\cdot | \tilde{\mathbf{Z}}_k^i)$ 
    set  $W_{k+1}^i = \tilde{W}_k^i$  and  $\tilde{W}_{k+1}^i = \frac{W_{k+1}^i G_{k+1}(\mathbf{Z}_{k+1}^i)}{\sum_j W_{k+1}^j G_{k+1}(\mathbf{Z}_{k+1}^j)}$ 
  if  $\forall j, \tilde{W}_{k+1}^j = 0$  then
     $\forall q > k$ , set  $\eta_q^N = \tilde{\eta}_q^N = 0$  and Stop
  else
     $k := k + 1$ 

```

Table 2.4 – SMC algorithm with adaptive resampling steps

expression of the variance of the SMC in the article [22] consider that the steps with a resampling are chosen deterministically, because the authors identify the variance of the SMC estimator with the variance of an IPS estimator for which the resampling steps are tuned to match the one selected by the SMC. To our knowledge we have not found a justification for this simplification.

Alternative resampling schemes

Note that in the presented algorithm one can use alternative strategies to select high potential trajectories. Here, the presented algorithms include a standard multinomial resampling procedure, but one can also use residual resampling or stratified resampling without altering the properties of the estimator. Empirical results suggest that these alternative resampling schemes yield estimations with smaller variances [25, 30]. There are also theoretical results on the higher performance of stratified resampling scheme compared to the classical multinomial scheme [gerber2017negative]. In [gerber2017negative] the authors also propose a new SSP resampling scheme for which the algorithm converges and is as efficient as the systematic resampling.

Other improvements

MCMC steps with invariant distribution $\tilde{\eta}_k$ can also be included in the algorithm after the resampling step. Some adaptations of the algorithm for parallel implementations have

also been studied in [52]. As the Feymann-kac model is used in IPS and SMC methods, any improvement for the SMC in the literature, might be of interest for the IPS. As the use of SMC for filtering or for Bayesian algorithms have also been an active subject of research in the last decades, we encourage the reader that would want go further in the improvement of the IPS to search the filtering and Bayesian literature where there might be some improvement that could be transposed for the IPS. Indeed, the literature is quite abundant on these subjects.

PART II

Importance Sampling with PDMPs

Remember that \mathbf{Z}_{t_f} is a trajectory of the state of the system on an interval $[0, t_f]$, and that this trajectory is modeled using a PDMP. For $h \in \mathcal{M}(\mathbf{E}_{t_f})$ we define

$$p = \mathbb{E}\left[h(\mathbf{Z}_{t_f})|Z_0 = z_0\right], \quad (2.84)$$

in this part we focus on the estimation of p with the importance sampling method.

Let \mathcal{D} be the set of the trajectories involving a system failure and that are defined on the interval defined on $[0, t_f]$. For reliability assessment purposes, we want to estimate the probability that the system fails before t_f , therefore we are especially interested in the case $h = \mathbb{1}_{\mathcal{D}}$. Indeed in that situation p is the probability of failure and

$$p = \mathbb{E}\left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f})|Z_0 = z_0\right] = \mathbb{P}\left(\mathbf{Z}_{t_f} \in \mathcal{D}|Z_0 = z_0\right). \quad (2.85)$$

In this context, importance sampling would consist in simulating from a more fragile system, while weighting the simulation outputs by the appropriate likelihood ratios. If the trajectories $\mathbf{Z}_{t_f}^{1'}$, $\mathbf{Z}_{t_f}^{2'}$, \dots , $\mathbf{Z}_{t_f}^{N'}$ have been generated from a more fragile system, an importance sampling estimator would have the following form:

$$\hat{p}_{IS} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{\mathcal{D}}\left(\mathbf{Z}_{t_f}^{i'}\right) \frac{f\left(\mathbf{Z}_{t_f}^{i'}\right)}{g\left(\mathbf{Z}_{t_f}^{i'}\right)} \quad (2.86)$$

where $\frac{f\left(\mathbf{Z}_{t_f}^{i'}\right)}{g\left(\mathbf{Z}_{t_f}^{i'}\right)}$ is the appropriate likelihood ratio. But as the variables of interest, i.e. the $\mathbf{Z}_{t_f}^{i'}$'s, are now trajectories of a PDMP, in order to use equation (2.86), we first need to define what is the likelihood ratio for these trajectories of PDMP.

The definition of these likelihood ratios for trajectories of PDMP, requires to define a reference measure so that we can define the densities involved in the likelihood ratio. The existence of such a reference measure for PDMP trajectories is crucial, because it preconditions the existence of the likelihood ratio needed to apply the importance sampling method. In simple cases of dynamical importance sampling, this issue of the reference measure is often eluded, because the reference measure has an obvious form: it is often a product of Lebesgue measures, or a product of discrete measures, but PDMPs are very degenerate processes for which the existence of a reference measure needs to be proven.

Chapter 1 shows that for a PDMP with boundaries, the time between two consecutive jumps can be a hybrid random variable. Such hybrid random variable has a reference measure which is a mixture of Dirac and Lebesgue measures. Secondly, these hybrid jumps may occur multiple times and in a nested way in the law of the trajectory of PDMP. With these mixtures of Dirac and Lebesgue measures involved, the existence of a sigma-finite reference measure on the trajectory space is not obvious, yet it is mandatory to properly define the density of a trajectory.

Chapter 3 therefore investigates the existence of such a reference measure, and introduces the theoretical foundations needed for applying importance sampling on trajectories of PDMP.

Note that the interest of this work on the reference measure and the densities is not purely theoretical, as knowing the reference measure and densities also allows to identify what are the possible importance processes, which is an important practical issue. Indeed, the choice of the importance process is paramount as it determines the variance reduction, and it is important to determine which parameters can be tuned to obtain an efficient important process.

In our search for more efficient importance processes, we give a specific attention to the optimal importance process, in the hope that with better information on this optimal process we can design importance processes closer to the optimal one, and therefore that are more efficient. We find the characteristics of the optimal process in Chapter 4, along with a convenient way to specify an importance process inspired from the optimal process.

Chapter 5 is then dedicated to the application of the method on two examples. The first example is a toy system, and the second example is inspired from a real industrial system.

The main results of this part are also presented in the article [15].

Theoretical foundation for Importance sampling on PDMP

3.1 Prerequisite for importance sampling

Suppose that ζ_{t_f} is a reference measure for the law of \mathbf{Z}_{t_f} , and that we denote by f the density of \mathbf{Z}_{t_f} with respect to ζ_{t_f} , and by g the density of an importance process \mathbf{Z}'_{t_f} with respect to ζ_{t_f} . If ζ_{t_f} exists, and f and g satisfy $\in \mathbf{E}_{t_f}$, $h(\mathbf{z})f(\mathbf{z}) \neq 0 \Rightarrow g(\mathbf{z}) \neq 0$, for ζ_{t_f} -almost every z , then we can write:

$$p = \mathbb{E}_f \left[h(\mathbf{Z}_{t_f}) \right] = \int_{\mathbf{E}_{t_f}} h(\mathbf{z}) f(\mathbf{z}) d\zeta_{t_f}(\mathbf{z}) = \int_{\text{supp}(h)} h(\mathbf{z}) \frac{f(\mathbf{z})}{g(\mathbf{z})} g(\mathbf{z}) d\zeta_{t_f}(\mathbf{z}) = \mathbb{E}_g \left[h(\mathbf{Z}_{t_f}) \frac{f(\mathbf{Z}_{t_f})}{g(\mathbf{Z}_{t_f})} \right]. \quad (3.1)$$

If $(\mathbf{Z}_{t_f}^1, \dots, \mathbf{Z}_{t_f}^N)$ is a sample of independent trajectories simulated according to an importance process with density g , then p can be estimated without bias by:

$$\hat{p}_{IS} = \frac{1}{N} \sum_{i=1}^N h(\mathbf{Z}_{t_f}^i) \frac{f(\mathbf{Z}_{t_f}^i)}{g(\mathbf{Z}_{t_f}^i)} \quad \text{with} \quad \text{Var}(\hat{p}_{IS}) = \frac{\mathbb{E}_f \left[h(\mathbf{Z}_{t_f})^2 \frac{f(\mathbf{Z}_{t_f})}{g(\mathbf{Z}_{t_f})} \right] - p^2}{N}. \quad (3.2)$$

When $\mathbb{E}_f \left[h(\mathbf{Z}_{t_f})^2 \frac{f(\mathbf{Z}_{t_f})}{g(\mathbf{Z}_{t_f})} \right] < \infty$ and the conditions above are verified, we have a central limit theorem on \hat{p}_g :

$$\sqrt{N}(\hat{p}_g - p) \longrightarrow \mathcal{N}(0, \sigma_{IS}^2) \quad \text{where} \quad \sigma_{IS}^2 = \mathbb{E}_f \left[h(\mathbf{Z}_{t_f})^2 \frac{f(\mathbf{Z}_{t_f})}{g(\mathbf{Z}_{t_f})} \right] - p^2. \quad (3.3)$$

Thus the use of importance sampling on PDMP trajectories requires the following three conditions:

- (C1) We have a measure ζ_{t_f} on the trajectory space, and the trajectory \mathbf{Z}_{t_f} of the system state has density f with respect to ζ_{t_f} ,
- (C2) We are able to simulate trajectories according to an importance process \mathbf{Z}'_{t_f} which has density g with respect to ζ_{t_f} on $\text{supp}(h)$ (note $\text{supp}(h) = \mathcal{D}$ when $h = \mathbb{1}_{\mathcal{D}}$), and such that $\mathbb{E}_f \left[h(\mathbf{Z}_{t_f})^2 \frac{f(\mathbf{Z}_{t_f})}{g(\mathbf{Z}_{t_f})} \right] < \infty$,
- (C3) For ζ_{t_f} -almost all $\mathbf{z} \in \mathbf{E}_{t_f}$, we have $h(\mathbf{z})f(\mathbf{z}) \neq 0 \Rightarrow g(\mathbf{z}) \neq 0$

Many authors have used importance sampling on particular cases of PDMP sometimes without noting they were PDMPs, see [36, 37, 39, 41]. Sometimes, the authors using PDMPs avoid considering automatic control mechanisms which activate and deactivate components depending on the values of physical variables. Such automatic control mechanisms play an important role in power generation systems, and therefore they can not be avoided in our case. Also, the modeling of control mechanisms implies to work with a special kind of PDMPs, which are the PDMPs with boundaries. These PDMP are typically the kind for which the reference measure is complex. In [41, 44], importance sampling is used on PDMP with boundaries but the reference measure is not clearly identified.

In order to identify a reference measure we start by studying the law of the trajectories.

3.1.1 The law of the trajectories

In order to build the law of the trajectories, we use the fact that a trajectory can be summed up by its skeleton. Remember that we denote by Θ_{t_f} the map that changes \mathbf{Z}_{t_f} into its skeleton $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$. The law of \mathbf{Z}_{t_f} can be defined as the image law of $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$ through Θ_{t_f} , but the law of the skeleton is yet to be defined.

We can get the law of $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$, by using the dependencies between its coordinates. Indeed the equation (1.58) provides the law of T_k knowing Z_{S_k} , as it gives

$$\mathbb{P}(T_k \leq t \mid Z_{S_k} = z) = \begin{cases} 1 - \exp\left[-\int_0^t \lambda_z(u) du\right] & \text{if } t < t_z^*, \\ 1 & \text{if } t \geq t_z^*, \end{cases} \quad (3.4)$$

and the equation (1.54) provides the law of $Z_{S_{k+1}}$ knowing (Z_{S_k}, T_k) , as

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}\left(Z_{S_{k+1}} \in B \mid Z_{S_{k+1}}^- = \Phi_{Z_{S_k}}(T_k)\right) = \mathcal{K}_{\Phi_{Z_{S_k}}(T_k)}(B). \quad (3.5)$$

Let μ_z be the measure on $(\mathbb{R}^+, \mathcal{B}(\mathbb{R}^+))$ defined by

$$\forall B \in \mathcal{B}(\mathbb{R}^+), \quad \mu_z(B) = \text{leb}(B \cap (0, t_z^*)) + \delta_{t_z^*}(B), \quad (3.6)$$

where $\text{leb}(\cdot)$ corresponds to the Lebesgue measure. We can define the density of T_k knowing Z_{S_k} with respect to $\mu_{Z_{S_k}}$ by

$$f_{T_k \mid Z_{S_k}=z}(u) = \left(\lambda_z(u)\right)^{\mathbb{1}_{u < t_z^*}} \exp\left[-\Lambda_z(u)\right], \quad (3.7)$$

as

$$\forall t \in [0, t_z^*], \quad \mathbb{P}(T_k \leq t \mid Z_{S_k} = z) = \int_0^t \left(\lambda_z(u) \right)^{\mathbb{1}_{u < t_z^*}} \exp \left[-\Lambda_z(u) \right] d\mu_z(u). \quad (3.8)$$

The density of $Z_{S_{k+1}}$ knowing (Z_{S_k}, T_k) with respect to $\nu_{Z_{S_{k+1}}^-}$ has already been defined by

$$f_{Z_{S_{k+1}} \mid Z_{S_k}, T_k}(z^+) = K_{\Phi_{Z_{S_k}}(T_k)}(z^+), \quad (3.9)$$

as we have:

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}\left(Z_{S_{k+1}} \in B \mid Z_{S_{k+1}}^- = \Phi_{Z_{S_k}}(T_k)\right) = \int_B K_{Z_{S_{k+1}}^-}(z^+) d\nu_{Z_{S_{k+1}}^-}(z^+). \quad (3.10)$$

Using the Markov structure of the sequence $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$, the law of $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$ can be expressed as an integral of the product of the conditional densities given by (3.7) and (3.9).

Definition 27. We define the σ -algebra \mathcal{S}_t , on the set of the possible values of $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_{t_f})}$ such that $\sum_{i=0}^{n(\mathbf{Z}_{t_f})} T_i = t$, as the σ -algebra generated by the sets in $\bigcup_{n \in \mathbb{N}^*} \mathcal{B}\left(\left\{(z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = t\right\}\right)$, where $\mathcal{B}(\cdot)$ indicates the Borelians of a set:

$$\mathcal{S}_t = \sigma \left(\bigcup_{n \in \mathbb{N}^*} \mathcal{B}\left(\left\{(z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = t\right\}\right) \right) \quad (3.11)$$

Similarly for an interval I of length $l > 0$, we define the σ -algebra \mathcal{S}_I on the set of the possible values of $(Z_{S_k}, T_k)_{k \leq n(\mathbf{Z}_I)}$ such that $\sum_{i=0}^{n(\mathbf{Z}_I)} T_i = l$ as the σ -algebra generated by the sets in $\bigcup_{n \in \mathbb{N}^*} \mathcal{B}\left(\left\{(z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = l\right\}\right)$, where $\mathcal{B}(\cdot)$ indicates the Borelians of a set:

$$\mathcal{S}_I = \sigma \left(\bigcup_{n \in \mathbb{N}^*} \mathcal{B}\left(\left\{(z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = l\right\}\right) \right) \quad (3.12)$$

Definition 28. The law of the trajectories is defined as follows, for $B \in \mathcal{S}_{t_f}$, if A_n is the

set of skeletons with n jumps:

$$\begin{aligned}
 \mathbb{P}_{z_o} \left(\mathbf{Z} \in \Theta_{t_f}^{-1}(B) \right) &= \mathbb{P}_{z_o} \left(\mathbf{Z} \in \Theta_{t_f}^{-1} \left(\bigcup_{n \in \mathbb{N}} B \cap A_n \right) \right) \\
 &= \sum_{n \in \mathbb{N}} \mathbb{P}_{z_o} \left(\mathbf{Z} \in \Theta_{t_f}^{-1}(B \cap A_n) \right) \\
 &= \sum_{n \in \mathbb{N}} \int_{B \cap A_n} \prod_{k=0}^n \left(\lambda_{z_k}(t_k) \right)^{\mathbb{1}_{t_k < t_k^*}} \exp \left[-\Lambda_{z_k}(t_k) \right] \prod_{k=1}^n K_{z_k^-}(z_k) \\
 &\quad \times d\delta_{t_n^*}^*(t_n) d\nu_{z_n^-}(z_n) d\mu_{t_{z_{n-1}}^*}^*(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_o}^*}^*(t_0)
 \end{aligned} \tag{3.13}$$

$$\begin{aligned}
 \mathbb{P}_{z_o} \left(\mathbf{Z}_{t_f} \in \Theta_{t_f}^{-1}(B) \right) &= \int_B \prod_{k=0}^n \left(\lambda_{z_k}(t_k) \right)^{\mathbb{1}_{t_k < t_k^*}} \exp \left[-\Lambda_{z_k}(t_k) \right] \prod_{k=1}^n K_{z_k^-}(z_k) \\
 &\quad \times d\delta_{t_n^*}^*(t_n) d\nu_{z_n^-}(z_n) d\mu_{t_{z_{n-1}}^*}^*(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_o}^*}^*(t_0),
 \end{aligned} \tag{3.14}$$

where $z_j^- = \Phi_{z_{j-1}}(t_{j-1})$, and $t_n^* = t_f - \sum_{i=0}^{n-1} t_i$.

Note that with our construction, this is a probability law on the space of the trajectories in $\Theta_{t_f}^{-1} \left(\bigcup_{n \in \mathbb{N}^*} \left\{ (z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n \mid \sum_{i=0}^n t_i = t_f \right\} \right)$, not on the set of all the trajectories with values in E .

3.1.2 The dominant measure and the density

Definition 29. For a time $t_f > 0$, we define the measure ζ_{t_f} on $\Theta_{t_f}^{-1}(\mathcal{S}_{t_f})$ so that

$$\zeta_{t_f}(\Theta_{t_f}^{-1}(B)) = \int_{(z_k, t_k)_{k \leq n} \in B} d\delta_{t_n^*}^*(t_n) d\nu_{z_n^-}(z_n) d\mu_{t_{z_{n-1}}^*}^*(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_o}^*}^*(t_0). \tag{3.15}$$

The equation (3.14) shows that ζ_{t_f} is a reference measure for the law of trajectories.

Theorem 16. *If $\exists C > 0, \forall z \in \bar{E}, \nu_z(E) < C$ and $t_f < \infty$, then ζ_{t_f} is a σ -finite measure. By Radon-Nikodym theorem, the density of a trajectory $\mathbf{z} = \Theta_{t_f}^{-1}((z_0, t_0), \dots, (z_n, t_n))$ with respect to the measure ζ_{t_f} is*

$$f(\mathbf{z}) = \prod_{k=0}^n \left(\lambda_{z_k}(t_k) \right)^{\mathbb{1}_{t_k < t_k^*}} \exp \left[-\Lambda_{z_k}(t_k) \right] \prod_{k=1}^n K_{z_k^-}(z_k). \tag{3.16}$$

Proof. Let $B_n = \left\{ (z_{s_k}, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = t_f \right\}$. Then $\Theta_{t_f}^{-1}(B_n)$ is the set of possible trajectories with n jumps, and the sets B_n for $n \in \mathbb{N}^*$ form a partition of the set

of all possible trajectories. Note that $B_n \subseteq (E \times [0, t_f])^n$, so

$$\begin{aligned} \zeta_{t_f}(\Theta_{t_f}^{-1}(B_n)) &\leq \zeta_{t_f}(\Theta_{t_f}^{-1}((E \times [0, t_f])^n)) \\ &\leq \int_{(E \times [0, t_f])^n} d\delta_{t_n}^*(t_n) d\nu_{z_n^-}(z_n) d\mu_{t_{z_{n-1}}^*}(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_0}^*}(t_0) \end{aligned}$$

We suppose that the ν_{z^-} are bounded, $\exists M > 0, \forall z^- \in \bar{E}, \nu_{z^-}(E) < M$. Under this assumption, we have:

$$\begin{aligned} \zeta_{t_f}(\Theta_{t_f}^{-1}(B_n)) &\leq M \int_{(E \times [0, t_f])^{n-1}} d\mu_{t_{z_{n-1}}^*}(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_0}^*}(t_0) \\ &\leq M \int_{(E \times [0, t_f])^{n-2}} \int_E \int_{[0, t_f]} d\mu_{t_{z_{n-1}}^*}(t_{n-1}) d\nu_{z_{n-1}^-}(z_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_0}^*}(t_0) \\ &\leq M(t_f + 1) \int_{(E \times [0, t_f])^{n-2}} \int_E d\nu_{z_{n-1}^-}(z_{n-1}) d\mu_{t_{z_{n-2}}^*}(t_{n-2}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_0}^*}(t_0) \\ &\leq M^2(t_f + 1) \int_{(E \times [0, t_f])^{n-2}} d\mu_{t_{z_{n-2}}^*}(t_{n-2}) d\nu_{z_{n-2}^-}(z_{n-2}) \dots d\nu_{z_1^-}(z_1) d\mu_{t_{z_0}^*}(t_0) \end{aligned}$$

By recurrence we get that $\zeta_{t_f}(\Theta_{t_f}^{-1}(B_n)) \leq M^n(t_f + 1)^n$, which proves that ζ_{t_f} is σ -finite.

The expression of the density then directly follows from equation (3.14). \square

Note that it is always possible to choose the measures ν_{z^-} so they are all bounded by the same constant. Indeed the transition Kernel is itself bounded by 1, as it is a probability measure. So, to get a measure ζ_{t_f} that is σ -finite, we can simply take the measures ν equal to the transition Kernel, so the densities are properly defined whenever the observation time t_f is finite.

The practical and optimal importance processes

4.1 Admissible importance processes

Recall that an admissible importance process is any process whose law is absolutely continuous with respect to ζ_{t_f} (condition C2), and which has a density g with respect to ζ_{t_f} satisfying $\forall \mathbf{z} \in \text{supp}(h)$, $h(\mathbf{z})f(\mathbf{z}) \neq 0 \Rightarrow g(\mathbf{z}) \neq 0$ ζ_{t_f} -almost everywhere (condition C3). In this section, we try to clarify the previous statement, and we try to identify to which extent we can modify the original process to obtain an admissible importance process. Throughout the rest of the thesis we denote the elements relative to this importance process with a $'$, except for its density that is denoted by g .

Our first remark is that condition C2 implies that the realizations of the importance process must satisfy equation (1.47). Indeed, the measure ζ_{t_f} involves the transformation Θ which uses the equation (1.47) to rebuild a trajectory from a skeleton. Consequently, the importance process has to piecewisely follow the same flows as the original process. The importance process is similar to the original one in the sense that it jumps to a new state each time there is a change in the flow. To ensure condition C2, the law of the T'_k has to be dominated by $\mu_{Z'_{S'_k}}$, and the law of $Z'_{S'_{k+1}}$ has to be dominated by $\nu_{Z'_{S'_k}}$. This means that the boundaries of the Ω_m 's and the set of the possible arrivals of a jump remain unchanged. So the modification of the original process focuses on the timing and nature of changes of modes, i.e. the laws of the jumps.

To generate an importance process, we keep generating trajectories by successively generating the arrival state of a jump ($Z'_{S'_k}$) and the time until the next jump (T'_k). As there is no requirement for the importance process to be Markovian, nor homogeneous, we consider that the law of a point of the trajectory Z'_t can depend on the past values of states, and the current time t . For all $t \in [0, t_f]$, we let $\mathbf{Z}'_t = (Z'_s)_{s \in [0, t]}$ and $\mathbf{Z}'_t^- = ((Z'_s)_{s \in [0, t)}, Z'_t^-)$ where $Z'_t^- = \lim_{u \uparrow t} Z'_u$. If S is the time of the last jump before t , the piece of trajectory on the interval $[S, t)$ is deterministic because it follows the flow. Therefore having Z'_t that depends on $\mathbf{Z}'_t^- = (Z'_s)_{s \in [0, t)}$ and t is like having Z'_t that depends on \mathbf{Z}'_S and t , and so, the

law of T'_k depends on $\mathbf{Z}'_{S'_k}$ and S'_k . As S'_k is a stopping time, one can also show that the law of $Z'_{S'_k}$ depends on $\mathbf{Z}'_{S'_k}$ and S'_k . Denoting by $\lambda'_{\mathbf{z}_s, s}$ the intensity function associated to T'_k when $S'_k = s$ and $\mathbf{Z}'_{S'_k} = \mathbf{z}_s$, we have:

$$\forall t \in [0, t_{z_k}^*],$$

$$\mathbb{P}(T'_k \leq t | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s) = \int_{[0, t]} \left(\lambda'_{\mathbf{z}_s, s}(u) \right)^{\mathbb{1}_{u < t_{z_s}^*}} \exp \left[- \Lambda'_{\mathbf{z}_s, s}(u) \right] d\mu_{z_s}(u). \quad (4.1)$$

The density, with respect to μ_{z_s} , of T'_k knowing $\mathbf{Z}'_{S'_k} = \mathbf{z}_s$ and $S'_k = s$ satisfies

$$\forall u \in [0, t_{z_s}^*], \quad f'_{T'_k | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}(u) = \left(\lambda'_{\mathbf{z}_s, s}(u) \right)^{\mathbb{1}_{u < t_{z_s}^*}} \exp \left[- \Lambda'_{\mathbf{z}_s, s}(u) \right] \quad \mu_{z_s}\text{-almost everywhere.} \quad (4.2)$$

Denoting by $K'_{\mathbf{z}_s^-}$ the importance Kernel density of the k -th jump when $\mathbf{Z}'_{S'_k} = \mathbf{z}_s^-$, we have:

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}(Z'_{S'_k} \in B | \mathbf{Z}'_{S'_k} = \mathbf{z}_s^-, S'_k = s) = \int_B K'_{\mathbf{z}_s^-}(z) d\nu_{z_s^-}(z) \quad (4.3)$$

Notice that the intensity function $\lambda'_{\mathbf{z}_s, s}$ in equation (4.1) does not have to be of the form $\lambda' \circ \phi_{z_s}$, where λ' is a positive function on E . This means that at the time $S'_k + t$, the intensity does not depend only on the state $Z'_{S'_k + t}$ as it would be the case if \mathbf{Z}'_{t_f} were a PDMP. So, in the importance process, we consider that the intensity can depend on the arrival state of the last jump and on previous states $\mathbf{Z}'_{S'_k}$ and on the current time. Therefore the importance process can be seen as a piecewise deterministic process (PDP) which is not necessarily Markovian nor homogeneous.

Definition 30 (Deterministic extension of trajectories). For a trajectory $\mathbf{z}_{[a, b]} = (z_s)_{s \in [a, b]}$ and $u \in [0, t_{z_b}^*]$, we define $\Phi_{\mathbf{z}}(u)$ as the jump-less extension of $\mathbf{z}_{[a, b]}$ on the interval $[a, b + u]$ so that, if $\mathbf{z}_{[a, b + u]} = \Phi_{\mathbf{z}_{[a, b]}}(u)$, then $\forall v \leq u$, $z_{b+v} = \Phi_{z_b}(v)$.

Concerning condition C3, we notice the following necessary and sufficient condition:

Property 6. The condition C3 is satisfied if and only if $\forall s > 0$, $\mathbf{z}_s \in \mathbf{E}_s$, $\mathbf{z}_s^- \in \mathbf{E}_s^-$, and $t \in (0, t_{z_s}^*]$ we have:

$$\text{if } \left\{ \mathbb{E} \left[|h(\mathbf{Z}_{t_f})| \middle| \mathbf{Z}_{S_k} = \mathbf{z}_s, S_k = s \right] > 0 \text{ and } K_{z_s^-}(z_s) > 0 \right\}, \text{ then } K'_{\mathbf{z}_s^-}(z_s) > 0,$$

$$\text{and if } \left\{ \mathbb{E} \left[|h(\mathbf{Z}_{t_f})| \middle| \mathbf{Z}_{S_k + t^-} = \Phi_{\mathbf{z}_s}(t), S_k = s \right] > 0 \text{ and } \lambda_{z_s}(t) > 0 \right\}, \text{ then } \lambda'_{\mathbf{z}_s}(t) > 0.$$

Unfortunately with complex systems, the support of h can be very hard to manipulate, and we do not always know if $\mathbb{E} \left[|h(\mathbf{Z}_{t_f})| \middle| \mathbf{Z}_{S_k} = \mathbf{z}_s, S_k = s \right]$ or

$\mathbb{E}[|h(\mathbf{Z}_{t_f})| | \mathbf{Z}_{S_k+t^-} = \Phi_{\mathbf{z}_s}(t), S_k = s]$ are positive. So in practice we often only use the following sufficient condition which states that for almost any $z_s \in E$, and $z_s^- \in \bar{E}$, and $t \in (0, t_s^*]$:

$$\begin{aligned} K_{z_s^-}(z_s) > 0 &\Rightarrow K'_{z_s^-}(z_s) > 0 \\ \lambda_{z_s}(t) > 0 &\Rightarrow \lambda'_{z_s}(t) > 0. \end{aligned}$$

4.2 A way to build an optimal importance process

Definition 31. Let U^* be the function defined on $\cup_{s \in [0, t_f]} \mathbf{E}_s \times \{s\}$ by:

$$U^*(\mathbf{z}, s) = \mathbb{E}[|h(\mathbf{Z}_{t_f})| | \mathbf{Z}_s = \mathbf{z}]. \quad (4.4)$$

Definition 32. Let U^- be the function defined on $\cup_{s \in [0, t_f]} \mathbf{E}_s^- \times \{s\}$ by:

$$U^-(\mathbf{z}_s^-, s) = \int_E U^*(\mathbf{z}_s, s) K_{z_s^-}(z_s) d\nu_{z_s^-}(z_s), \text{ where } \mathbf{z}_s = (\mathbf{z}_s^-, z_s). \quad (4.5)$$

These quantities play an important role in the following sections, as shown by the next theorem:

Theorem 17. *The densities*

$$g_{T_k^* | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s+u)}{U^*(\mathbf{z}_s, s)} f_{T_k | Z_{S_k} = z_s}(u) \quad (4.6)$$

and kernel densities

$$K_{\mathbf{z}_s^-, s_k}^*(z_s) = \frac{U^*(\mathbf{z}_s, s)}{U^-(\mathbf{z}_s^-, s)} K_{z_s^-}(z_s) \quad (4.7)$$

correspond to the jump densities and the transition kernel densities of an optimal importance process. Moreover these densities are the only ones that can be associated with an optimal process with density g^* defined by $g^*(\mathbf{z}) = \frac{|h(\mathbf{z})|f(\mathbf{z})}{p}$.

Proof. Assume the trajectory $\mathbf{z} = \Theta((z_{s_0}, t_0), \dots, (z_{s_n}, t_n))$ has been simulated with (4.6) and (4.7). (Remember we take the convention that $s_{n+1} = t_f$.) Then the density of the trajectory $g(\mathbf{z})$ with respect to ζ_{t_f} is defined by:

$$g(\mathbf{z}) = \prod_{k=0}^n g_{T_k^* | \mathbf{Z}'_{S'_k} = \mathbf{z}_{s_k}, S'_k = s_k}(t_k) \prod_{k=1}^n K_{\mathbf{z}_{s_k}^-}^*(z_{s_k})$$

So it verifies:

$$\begin{aligned}
 g(\mathbf{z}) &= \prod_{k=0}^n \frac{U^-(\Phi_{\mathbf{z}_{s_k}}(t_k), s_k + t)}{U^*(\mathbf{z}_{s_k}, s_k)} \prod_{k=1}^n \frac{U^*(\mathbf{z}_{s_k}, s_k)}{U^-(\mathbf{z}_{s_k}^-, s_k)} \prod_{k=0}^n f_{T_k|Z_{S_k}=z_{s_k}}(t_k) \prod_{k=1}^n K_{z_{s_k}^-}(z_{s_k}) \\
 &= \prod_{k=0}^n \frac{U^-(\mathbf{z}_{s_{k+1}}^-, s_{k+1})}{U^*(\mathbf{z}_{s_{k+1}}, s_{k+1})} \prod_{k=0}^{n-1} \frac{U^*(\mathbf{z}_{s_{k+1}}, s_{k+1})}{U^-(\mathbf{z}_{s_{k+1}}^-, s_{k+1})} f(\mathbf{z}) \\
 &= \frac{U^-(\mathbf{z}_{s_{n+1}}^-, s_{n+1})}{U^*(\mathbf{z}_{s_0}, s_0)} f(\mathbf{z}) = \frac{|h(\mathbf{z})|f(\mathbf{z})}{\mathbb{E}[|h(\mathbf{Z}_{t_f})||Z_0 = z_{s_0}]} = g^*(\mathbf{z}),
 \end{aligned}$$

where $g^*(\mathbf{z})$ is the optimal density density. This proves the first part of the theorem.

Assume that $g'_{\mathbf{z}_{s_k}, s_k}$ and $K'_{\mathbf{z}_{s_k}^-, s_k}$ are respectively jump densities and transition kernel densities such that for any trajectory $\mathbf{z} = \Theta((z_{s_0}, t_0), \dots, (z_{s_n}, t_n))$ we have

$$\begin{aligned}
 &\prod_{k=0}^n g'_{\mathbf{z}_{s_k}, s_k}(t_k) \prod_{k=1}^n K'_{\mathbf{z}_{s_k}^-, s_k}(z_k) = g^*(\mathbf{z}). \\
 \text{So } &\prod_{k=0}^n g'_{\mathbf{z}_{s_k}, s_k}(t_k) \prod_{k=1}^n K'_{\mathbf{z}_{s_k}^-, s_k}(z_k) = \prod_{k=0}^n g_{T'_k|Z'_{S'_k}=\mathbf{z}_{s_k}, S'_k=s_k}^*(t_k) \prod_{k=1}^n K_{\mathbf{z}_{s_k}^-, s_k}^*(z_k)
 \end{aligned}$$

As all these densities integrate to one, by integrating over $(z_{s_1}, t_1, z_{s_2}, t_2, \dots, z_n, t_n)$ on both sides, we obtain that

$$\forall s_0, z_{s_0}, t_0, \quad g'_{\mathbf{z}_{s_0}, s_0}(t_0) = g_{T'_0|Z'_{S'_0}=\mathbf{z}_{s_0}, S'_0=s_0}^*(t_0)$$

Then by integrating over $(t_{s_1}, z_{s_2}, t_2, \dots, z_{s_n}, t_n)$, we obtain that

$$\forall \mathbf{z}_{s_1}^- = \Phi_{\mathbf{z}_{s_0}}(t_0), s_1 = t_0 \quad K'_{\mathbf{z}_{s_1}^-, s_1}(z_{s_1}) = K_{\mathbf{z}_{s_1}^-, s_1}^*(z_{s_1}).$$

By recurrence of this reasoning, we can show that for any possible k -th jump the densities $g'_{\mathbf{z}_{s_k}, s_k}$ and $K'_{\mathbf{z}_{s_k}^-, s_k}$ are identical to $g_{T'_k|Z'_{S'_k}=\mathbf{z}_{s_k}, S'_k=s_k}^*$ and $K_{\mathbf{z}_{s_k}^-, s_k}^*$, which proves the uniqueness of the jump densities such that $g(\mathbf{z}) = g^*(\mathbf{z})$. \square

Equations (4.6) and (4.7) serve as a guide to build an importance process: one should try to specify densities as close as possible to these equations so as to get an estimator variance as close as possible to the optimal variance.

Also note that we could have used any function proportional to U^* instead of U^- , because the proximity function always appears in a ratio $\frac{U^*}{U^-}$ or $\frac{U^-}{U^*}$. Indeed $U^- = \int U^* d\mathcal{K}$ so a proportionality constant would be canceled out in the ratios.

4.3 Remarks on the optimal process

As we generally do not know the explicit forms of U^* and U^- , the construction of an importance process close to the optimal one is delicate. Nonetheless, the equations (4.6) and (4.7) can give us information on how to build an importance process in practice. In this section, we investigate the properties of the optimal importance process and of the function U^* with the aim of building a good and practical importance process.

For instance, we can get the expression of the jump rate of the optimal process. For the time of the k -th jump, by definition of the jump rate and knowing that $\mathbf{Z}'_{S'_k} = \mathbf{z}_s$ and that $S'_k = s$, we get:

$$\begin{aligned} \lambda_{\mathbf{z}_s, s}^*(u) &= \frac{g_{T'_k | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}^*(u)}{\int_{(u, t_z^*]} g_{T'_k | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}^*(v) dv}, \\ \Leftrightarrow \lambda_{\mathbf{z}_s, s}^*(u) &= \frac{U^-(\Phi_{\mathbf{z}_s}(u), s+u) \left(\lambda_{\mathbf{z}_s}(u) \right)^{\mathbb{1}_{u < t_z^*}} \exp \left[-\Lambda_{\mathbf{z}_s}(u) \right]}{\int_{(u, t_z^*]} U^-(\Phi_{\mathbf{z}_s}(v), s+v) \left(\lambda_{\mathbf{z}_s}(v) \right)^{\mathbb{1}_{v < t_z^*}} \exp \left[-\Lambda_{\mathbf{z}_s}(v) \right] d\mu_{\mathbf{z}_s}(v)}. \end{aligned} \quad (4.8)$$

Using some properties of U^* and (4.8) we can prove the following theorem:

Theorem 18. *The jump rate of the optimal importance process defined by the densities (4.6) and (4.7) verifies:*

$$\lambda_{\mathbf{z}_s, s}^*(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s+u)}{U^*(\Phi_{\mathbf{z}_s}(u), s+u)} \lambda_{\mathbf{z}_s}(u). \quad (4.9)$$

The proof is provided in appendix V.

Note that this expression (4.9) can be easily interpreted. $\lambda_{\mathbf{z}_s, s}^*(u)$ corresponds to the jump rate at the state $Z_{s+u} = \Phi_{\mathbf{z}_s}(u)$ when $\mathbf{Z}_{s+u} = \Phi_{\mathbf{z}_s}(u)$. $U^*(\Phi_{\mathbf{z}_s}(u), s+u)$ is the probability of generating a failing trajectory knowing $\mathbf{Z}_{s+u} = \Phi_{\mathbf{z}_s}(u)$ and knowing there is no jump at time $s+u$. $U^-(\Phi_{\mathbf{z}_s}(u), s)$ is the probability of generating a failing trajectory knowing there is a jump at time $s+u$ from the departure state is $Z_{s+u}^- = \Phi_{\mathbf{z}_s}(u)$ and knowing $\mathbf{Z}_{s+u}^- = \Phi_{\mathbf{z}_s}(u)$. So the ratio $\frac{U^-(\Phi_{\mathbf{z}_s}(u), s+u)}{U^*(\Phi_{\mathbf{z}_s}(u), s+u)}$ is the factor multiplying the expectation of h when there is a jump at time $s+u$. The expression indicates that, in order to reach the optimal variance, one should increase the original jump rate in the same proportion as a jump would increase the expectation of h . In the case $h = \mathbb{1}_{\mathcal{D}}$, the expression indicates that, in order to reach the optimal variance, one should increase the original jump rate in the same proportion as a jump would increase the probability of getting a failing trajectory.

Also note that equations (4.9) and (4.7) indicate that, once the region D has been

reached, the optimal process does not differ from the original process. Indeed if τ_D is the reaching time of the critical region D , then for $s \geq \tau_D$ we have for all trajectories \mathbf{z}_s and \mathbf{z}_s^- , $U^*(\mathbf{z}_s, s) = U^-(\mathbf{z}_s^-, s) = 1$ and so for $s \geq \tau_D$ we get $K_{\mathbf{z}_s^-, s}^* = K_{z_s^-}$, and for $s + u \geq \tau_D$ we get $\lambda_{\mathbf{z}_s, s}^*(u) = \lambda_{z_s}(u)$.

The theorem 18 is noteworthy, because in practice the law of the jump time is specified through the jump rate. Thus it can be used to specify the laws of the jump times of an importance process, as we will do in chapter 5.

As it plays an important role in the expression of the optimal process, we look for more information about the function U^* . We first notice that: if τ is a stopping time such that $t_f > \tau > s$, then

$$\begin{aligned} U^*(\mathbf{z}_s, s) &= \mathbb{E}\left[h(\mathbf{Z}_{t_f}) \mid \mathbf{Z}_s = \mathbf{z}_s\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_{t_f}) \mid \mathbf{Z}_\tau\right] \mid \mathbf{Z}_s = \mathbf{z}_s\right] \\ \text{and so } U^*(\mathbf{z}_s, s) &= \mathbb{E}\left[U^*(\mathbf{Z}_\tau, \tau) \mid Z_s = z\right]. \end{aligned} \quad (4.10)$$

Using (4.10) we can show the two following properties:

Theorem 19. *U^* is kernel invariant on boundaries:*

$$\forall z \in E, \quad \lim_{t \nearrow t_z^*} U^*(\Phi_{\mathbf{z}_s}(t), s + t) = U^-(\Phi_{\mathbf{z}_s}(t_z^*), s + t_z^*). \quad (4.11)$$

Theorem 20. *If $u \rightarrow U^-(\Phi_{\mathbf{z}_s}(u), s + u)$ and $u \rightarrow \lambda_z(u)$ are continuous almost everywhere on $[0, t_z^*)$, then almost everywhere U^* is differentiable along the flow, with:*

$$\frac{\partial U^*(\Phi_{\mathbf{z}_s}(v), s + v)}{\partial v} = \left(U^*(\Phi_{\mathbf{z}_s}(v), s + v) - U^-(\Phi_{\mathbf{z}_s}(v), s + v) \right) \lambda_z(v) \quad (4.12)$$

The theorems 19 and 20 can in fact be seen as forward Kolmogorov equations on U^* . A complete proof for these two theorems is in the appendix V.

Though we think these equations might be useful, we did not manage to use these equations to identify U^* . This, mainly because there are many functions that also satisfy these equations. Indeed, the proof in the appendix V also works for any function of the form $\mathbf{z}_s \rightarrow \mathbb{E}[\tilde{h}(\mathbf{Z}_{t_f}) \mid \mathbf{Z}_s = \mathbf{z}_s]$, where \tilde{h} is a measurable function on \mathbf{E}_{t_f} .

4.4 Practical importance processes for reliability assessment

We now focus on the reliability assessment case, and we set $h = \mathbb{1}_\emptyset$ until the end of part II. We have seen in section 4.1, that the admissible importance processes have jump densities that depend on the trajectory already followed (see (4.1) and (4.3)). In the case $h = \mathbb{1}_\emptyset$, we can restrict the search of an efficient importance process within a special class of processes without any loss in efficiency, because the optimal importance process with density $g^*(\mathbf{z}) = \frac{\mathbb{1}_\emptyset(\mathbf{z})f(\mathbf{z})}{p}$ belongs to this special class.

The processes of this class can be defined through the expressions (4.1) and (4.3) too but, they do not use all the information contained in \mathbf{z}_s and \mathbf{z}_s^- . In this special class the jump rates $\lambda'_{\mathbf{z}_s, s}(t)$ depend only on three variables which are: the arrival state of the last jump z_s , the time $s + t$, and the indicator $\mathbb{1}_{\tau_D \leq s+t}$ which tells if the system failure has already happened. The kernels $\mathcal{K}'_{\mathbf{z}_s^-}$ depend only on three variables, which are: the current departure state z_s^- , the time s left before t_f , and the indicator $\mathbb{1}_{\tau_D \leq s}$.

To ease the presentation of such jump rates and transition kernels densities, we slightly modify the state space by adding an active boundary at the boundary of D and we add a coordinate on the mode which indicates if the trajectory has already visited D . The state now becomes $Z = (X, (M, M_D))$ where $M_D = 0$ if D has not been visited, and 1 if it has. This way, for any time t we have $Z_t = (X_t, (M_t, \mathbb{1}_{\tau_D \leq t}))$. For instance, with the heated-room system the set of modes becomes $\mathbb{M} = \{ON, OFF, F\}^3 \times \{0, 1\}$. The Kernel density K_{Z^-} is unchanged when $M_D^- = M_D^+$, and is null when $M_D^- \neq M_D^+$, except at the boundary of D where $K_{(0, (F, F, F, 0))}(0, (F, F, F, 1)) = 1$.

The three variables that determine the jump rates and kernels densities of the processes of the special class can now be identified by the current state and the current time. Therefore, we now consider importance processes with jump rate $\lambda'_{z_k, s_k}(t)$ and transition kernel density $K'_{z_k^-, s_k}$. Such processes have the following laws of jump times and jump arrivals:

$$\begin{aligned} \forall t \in (0, t_{z_k}^*], \quad \mathbb{P}(T'_k \leq t | Z'_{S'_k} = z_k, S'_k = s_k) \\ = \int_{(0, t]} \left(\lambda'_{z_k, s_k}(u) \right)^{\mathbb{1}_{u < t_{z_k}^*}} \exp \left[- \Lambda'_{z_k, s_k}(u) \right] d\mu_{z_k}(u) \end{aligned} \quad (4.13)$$

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}(Z'_{S'_k} \in B | Z'_{S'_k} = z_k^-, S'_k = s_k) = \int_B K'_{z_k^-, s_k}(z) d\nu_{z_k^-}(z). \quad (4.14)$$

Note that the class of processes that can be defined by (4.13) and (4.14) is included in the class of admissible importance processes.

With this new definition of the states, the quantity $U^*(\mathbf{z}_s, s)$ does not depend on the whole trajectory \mathbf{z}_s but only on the last state z_s and the time s . Indeed, as Z_t is Markovian

we can write

$$U^*(\mathbf{z}_s, s) = \mathbb{E}\left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \mid \mathbf{Z}_s = \mathbf{z}_s\right] = \mathbb{E}\left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \mid Z_s = z_s\right]. \quad (4.15)$$

So for the case $h = \mathbb{1}_{\mathcal{D}}$, we also redefine U^* and U^- :

Definition 33. If $h = \mathbb{1}_{\mathcal{D}}$, we let U^* be the proximity function defined on $E \times [0, t_f]$ by:

$$U^*(z, s) = \mathbb{E}\left[|h(\mathbf{Z}_{t_f})| \mid Z_s = z\right]. \quad (4.16)$$

Definition 34. For the case $h = \mathbb{1}_{\mathcal{D}}$, we let U^- be the function defined on $\bar{E} \times [0, t_f]$ by:

$$U^-(z^-, s) = \int_E U^*(z, s) K_{z^-}(z) d\nu_{z^-}(z). \quad (4.17)$$

In the case $h = \mathbb{1}_{\mathcal{D}}$, we call U^* the proximity function, because in that case, we have $U^*(z, s) = \mathbb{E}[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \mid Z_s = z]$: so the quantity $U^*(z, s)$ measures the proximity between the situation $Z_s = z$ and a system failure before t_f . The quantity $U^-(z^-, s)$ measures the chances of having a system failure before t_f knowing the system is jumping from the state z^- at time s .

Definition 35. If $h = \mathbb{1}_{\mathcal{D}}$, the optimal jump rates and jump kernel densities are such that:

$$\forall z \in E, s \in [0, t_f], \quad \lambda_{z,s}^*(u) = \frac{U^-(\Phi_{z_s}(u), s+u)}{U^*(\Phi_{z_s}(u), s+u)} \lambda_z(u) \quad (4.18)$$

$$\forall z^- \in \bar{E}, z \in E, s \in [0, t_f], \quad K_{z^-,s}^*(z) = \frac{U^*(z, s)}{U^-(z^-, s)} K_{z^-}(z) \quad (4.19)$$

4.5 A parametric family of importance processes for reliability assessment

In order to find an importance process that gives a good variance reduction, we usually restrict the search within a parametric family of importance densities. Then we rely on optimization routines to find the parameters yielding the best variance reduction. Here, we propose to use a parametric approximation of $U^*(z, s)$, or to any function proportional to $U^*(z, s)$, and then to combine it with equations (4.18) and (4.19) to get the form of the importance kernels and of the importance intensities. If we denote $U_\alpha(z, s)$ our approximation of $U^*(\mathbf{z}_s, s)$, and set $U_\alpha^-(z^-, s) = \int_E U_\alpha(z, s) K_{z^-}(z) d\nu_{z^-}(z)$, then the

corresponding importance intensities and Kernels are given by:

$$\lambda'_{z,s}(u) = \frac{U_{\alpha}^{-}(\Phi_z(u), s+u)}{U_{\alpha}(\Phi_z(u), s+u)} \lambda_z(u) , \quad (4.20)$$

$$K'_{z^-,s}(z) = \frac{U_{\alpha}(z, s)}{U_{\alpha}^{-}(z^-, s)} K_{z^-}(z) . \quad (4.21)$$

With these settings and notations, condition (C3) can be expressed as:

$$\left\{ U^*(z_k, s_k) > 0 \text{ and } K_{z_k^-}(z_k) > 0 \right\} \Rightarrow U_{\alpha}(z_k, s_k) > 0$$

$$\left\{ U^{-}(\Phi_{z_k}(t), s_k + t) > 0 \text{ and } \lambda_{z_k}(t) > 0 \right\} \Rightarrow U_{\alpha}^{-}(\Phi_{z_k}(t), s_k + t) > 0,$$

for any $z_k \in E$, and $z_k^- \in \bar{E}$, and $t \in (0, t_{z_k}^*]$. It is therefore satisfied if we take U_{α} positive everywhere for instance.

Here we switch the problem of setting a density g close to g^* by finding λ' and K' , to the problem of finding a surface U_{α} on $E \times \mathbb{R}^+$ close to the surface U^* .

Note that this way of building a parametric family of importance processes can be applied to any kind of system, though the form of U_{α} may have to be adapted from case to case. Indeed, we expect the form of U^* to depend on the configuration of the system and so does the form of the U_{α} 's.

We could also have plugged the approximations U_{α} and U_{α}^{-} into (4.6), rather than into (4.9), but the option we have chosen is in fact more convenient and computationally more efficient. With the equation (4.20), we pass through the intensity, so the associated densities of the T'_k 's automatically integrate to 1. Conversely if we had passed through equation (4.6), setting

$$g_{T'_k|Z'_{S'_k}, S'_k=z, s}(u) \propto \frac{U_{\alpha}^{-}(\Phi_z(u), s+u)}{U_{\alpha}^*(z, s)} f_{T_k|Z_{S_k}=z}(u)$$

we would need, before simulating a realization of the T'_k , to renormalize this density so that it integrates to 1. As this renormalization requires to compute an integral, the use of (4.6) is more computationally demanding than the use of (4.20).

Applications to power generation systems

In this chapter we start by applying importance sampling on a small system which is the heated room system presented in section 1.3.4, then we present some heuristics to build families of parametric proximity functions for industrial systems, and lastly we apply importance sampling on the spent-fuel-pool system.

5.1 Application to the Heated-Room system

In this section we present how we build an importance process for the heated room system presented in section 1.3.4. Recall that, for $Z_s = z$, if T denotes the time between s and the next jump, the reference measure for the density of T is μ_z . This measure is such that

$$\forall B \in \mathcal{B}(\mathbb{R}^+), \mu_z(B) = \mu_{Leb}(B \cap [0, t_z^*]) + \delta_{t_z^*}(B) \quad (5.1)$$

where if $z = (x, m)$ then t_z^* is the time until we hit a boundary : $t_z^* = \inf\{s > 0, \Phi_z(s) \in E_m\}$. The reference measure for the arrival state, when the departure state is $z^- = (x^-, m^-)$, is denoted ν_{z^-} and we have

$$\forall B \in \mathcal{B}(E), \quad \mathcal{K}_{z^-}(B) = \int_B K_{z^-}(z) \nu_{z^-}(z). \quad (5.2)$$

In the heated-room system we have

$$\forall B \in \mathcal{B}(E), \quad \nu_{z^-}(B) = \sum_{m \in \mathbb{M} \setminus \{m^-\}} \delta_{(x^-, m)}(B). \quad (5.3)$$

5.1.1 A parametric family of importance processes for the Heated-Room system

In the heated-room system, the three heaters are identical and are in parallel redundancy, so we expect the probability $U^*(z, s) = \mathbb{E}[\mathbb{1}_{\mathcal{D}}(\mathbf{z}) | Z_s = z]$ to increase with the

number of failed heaters in the state z . Therefore, noting $b(z)$ the number of failed heaters in state z , we start by setting

$$U_\alpha(z, s) = H_\alpha(b(z)) Q(x, s) \quad (5.4)$$

where Q is a function of position and time, and H_α is a function on integers. We set $H_\alpha(0) = 1$. As we want $U_\alpha(z, s)$ to increase with $b(z)$, H_α has to be an increasing function.

If T denotes the time until the next jump after a time s , using (4.10) with $\tau = s + T$ we get:

$$U^*(z, s) = \mathbb{E}\left[U^*(Z_{s+T}, s + T) \mid Z_s = z\right]. \quad (5.5)$$

As the repair rates are larger than the failure rates by one order of magnitude in practice, when there is at least one failed heater, the probability of arriving in a more degraded state Z_T is much lower than the probability of having a repair. This last remark can actually be applied to any reliable industrial system (see for instance [17]). Ideally we would like U_α to mimic the property of U^* so we would like to have

$$U_\alpha(z, s) = \mathbb{E}\left[U_\alpha(Z_T, s + T) \mid Z_s = z\right] \quad (5.6)$$

which can be reformulated as :

$$H_\alpha(b(z)) = \sum_{m^+ \in \mathbb{M}} H_\alpha(b(x, m^+)) \int_{(0, t_z^*]} K_{\Phi_z(u)}((\phi_x^m(u), m^+)) w_z(u) d\mu_z(u) \quad (5.7)$$

where $w_z(u) = \frac{Q(\phi_x^m(u), s + u)}{Q(x, s)} \exp[-\Lambda_z(u)]$, and remember ϕ_x^m is the flow on the position not on the state. As a repair is much more likely than failure, if the transition from state $(\phi_x^m(u), m)$ to the state $(\phi_x^m(u), m^+)$ indexes a repair, then $K_{\Phi_z(u)}((\phi_x^m(u), m^+))$ is larger than if it had indexed a failure. So, (5.7) implies that, when $b(z) > 1$, the value of $H_\alpha(b(z))$ is closer from $H_\alpha(b(z) - 1)$ than from $H_\alpha(b(z) + 1)$. As H_α is supposed to be increasing, it must also be convex. So we propose that $H_\alpha(b(z)) = \exp[\alpha_1 b(z)^2]$, with $\alpha_1 > 0$. If, from a $Z_{s-} = \Phi_z(u)$, the transition j corresponds to a failure then we have:

$$\forall u \in [0, t_z^*), \quad \lambda'_{z,s}{}^j(u) = \lambda_z^j(u) \exp[\alpha_1(2b(z) + 1)] , \quad (5.8)$$

and if it corresponds to a repair then we have:

$$\forall u \in [0, t_z^*), \quad \lambda'_{z,s}{}^j(u) = \lambda_z^j(u) \exp[-\alpha_1(2b(z) - 1)] . \quad (5.9)$$

If J is the set of all transition 's indices, an j is the index of the transition from z to z^+ ,

the jump rate and the kernel density satisfy:

$$\forall u \in [0, t_z^*), \quad \lambda'_{z,s}(u) = \sum_{i \in J} \lambda'_{z,s}{}^i(u) \quad \text{and} \quad K'_{\Phi_z(u)^-,s}(z^+) = \frac{\lambda'_{z,s}{}^j(u)}{\sum_{i \in J} \lambda'_{z,s}{}^i(u)}. \quad (5.10)$$

We set the jump kernel such that its density satisfies for any departure state $z^- \in E$:

$$K'_{z^-}(z^+) = \frac{K_{z^-}(z^+) \exp[-\alpha_1 b(z^+)^2]}{\int_E K_{z^-}(z) \exp[-\alpha_1 b(z)^2] d\nu_{z^-}(z)}. \quad (5.11)$$

Note that plugging U_α into the equations (4.18) and (4.19) imposes some kind of symmetry in the biasing of failure and repair rates. It is especially visible in equations (5.8) and (5.9): On the one hand the failure rate associated to the transition from a state z^- to z^+ is multiplied by a factor $\exp[\alpha_1(2b(z^-) + 1)]$, and on the other hand the repair rate corresponding to the reversed transition (from state z^+ to state z^-) is divided by a factor $\exp[\alpha_1(2b(z^-) - 1)]$. The equations (4.18) and (4.19) not only imply that the failures should be enhanced and the repairs inhibited, but it also states that the magnitudes of the distortions should be reciprocal.

The square in H_α 's formula is introduced to strengthen the failure rates when the number of broken heaters gets larger. The idea is to shorten the duration where several heaters are simultaneously failed in the simulated trajectories. Indeed, as repair is faster than failure, the shorter are the durations with a failed heater the more likely is the trajectory. Increasing the failure rates with the number of broken heaters is a mean to simulate more trajectories in \mathcal{D} while maintaining the natural proportion between the likelihoods of the trajectories, which should decrease the variance.

As the failure on demand is likely to play an important role in the system failure, we choose to separate it from spontaneous failure in our parametrisation setting $U_\alpha((x_{min}, m), s) = \exp[-\alpha_2 b(z)^2] H_\alpha(x_{min}, s)$. This allows to better fit U_α to U^* . Under this assumption, the equation (4.21) implies that for $z^- = (x_{min}, m)$, the importance kernel takes this form:

$$K'_{z^-}(z^+) = \frac{K_{z^-}(z^+) \exp[-\alpha_2 b(z^+)^2]}{\int_E K_{z^-}(z) \exp[-\alpha_2 b(z)^2] d\nu_{z^-}(z)}. \quad (5.12)$$

5.1.2 Results

The Monte-Carlo simulations have been carried out using the Python library PyCATSHOO. (The flow functions ϕ_x^m were computed using a Runge-Kutta method of order 4 with a discretization step of 0.01. This discretization step is small enough so that reducing it further does not change the estimations.) As the Cross-Entropy method was not yet implemented in PyCATSHOO, we have used a specific Python code for the Cross-Entropy

and the importance sampling methods. The system parameters used in the simulation were the following ones: $x_{min} = 0.5$, $x_{max} = 5.5$, $x_e = -1.5$, $\beta_1 = 0.1$, $\beta_2 = 5$, $t_f = 100$. Trajectories were all initiated in the state $z_0 = (7.5, (OFF, OFF, OFF))$. The probability of having a system failure before t_f was estimated to $p = 1.29 \times 10^{-5}$ with an intensive Monte-Carlo estimation based on 10^8 runs.

	N_{sim}	\hat{p}	$\hat{\sigma}^2/N_{sim}$	IC $\times 10^5$	t_{sim}	\widehat{eff}
IS	10^3	1.28×10^{-5}	4.37×10^{-13}	[1.15, 1.41]	0.073 s	3.1×10^{10}
	10^4	1.273×10^{-5}	5.07×10^{-14}	[1.228, 1.317]	0.073 s	2.7×10^{10}
	10^5	1.289×10^{-5}	5.01×10^{-15}	[1.275, 1.303]	0.077 s	2.6×10^{10}
	10^6	1.288×10^{-5}	5.05×10^{-16}	[1.283, 1.292]	0.079 s	2.5×10^{10}
MC	10^6	0.4×10^{-5}	4.00×10^{-12}	[0.01, 0.79]	0.022 s	no convergence
	10^7	1.3×10^{-5}	1.28×10^{-12}	[1.07, 1.51]	0.022 s	3.5×10^6

Table 5.1 – Comparison between Monte-Carlo and importance sampling estimations

The values of the parameters selected by the cross-entropy method were $\alpha_1 \simeq 0.915$ and $\alpha_2 \simeq 1.197$, and for the first step, the approximation of the Kullback-Leibler divergence between g^* and g_α was obtained by simulating from a biased density with parameters $(0.5, 0.5)$. The whole cross-entropy method lasted approximately 9 minutes. Most of the running time was allocated to the optimization within each step of the cross-entropy, because each evaluation of the objective function and of its gradient was costly. In order to optimize the running time of the cross-entropy method, the size of the sample used for the approximations of the Kullback-Leibler divergence was set by simulating until we would get $n_{CE} = 100$ trajectories with a system failure. The number of $n_{CE} = 100$ roughly guaranties that the two first digits of the Kullback-Leibler divergences are identified by their approximations. For each of the three steps needed to select the parameters, samples of respectively 1970, 126, 127 trajectories were used.

A comparison between Monte-Carlo and the associated importance sampling estimates is presented in Table 5.1, where we display the number N_{sim} of simulations used for each method, the estimates \hat{p} of the probability, the associated empirical variances $\hat{\sigma}^2/N_{sim}$ and confidence intervals $IC = [\hat{p} - 1.96\sqrt{\hat{\sigma}^2/N_{sim}}, \hat{p} + 1.96\sqrt{\hat{\sigma}^2/N_{sim}}]$, and the mean time of a simulation t_{sim} in seconds. For 10^6 simulations the results show that the Monte-Carlo estimator has not converged yet, whereas the importance sampling estimate is very accurate. To compare the two methods we estimate the efficiency of their estimators when they have converged. The efficiency is defined by the ratio of the precision and the computational time:

$$eff = \frac{1}{\sigma^2/N_{sim}} \times \frac{1}{N_{sim}t_{sim}} = \frac{1}{\sigma^2t_{sim}}.$$

The efficiency can be interpreted as the contribution of a second of computation to the precision of the estimator. We estimate it by $\widehat{eff} = \frac{1}{\hat{\sigma}^2 t_{sim}}$. The results indicate that our importance sampling strategy is approximately 7 000 times more efficient than a Monte-Carlo method.

We also verify that the importance sampling estimations are asymptotically normally distributed. The asymptotic normality was not observed for $N_{sim} = 10^3$, but it was observed for larger sample sizes. For instance for $N_{sim} = 10^4$, the Figure 5.1 shows a normalized histogram on 100 estimations \hat{p}_{IS} that matches the normal density with mean p and with the empirical standard deviation of the 100 estimations. We also recorded

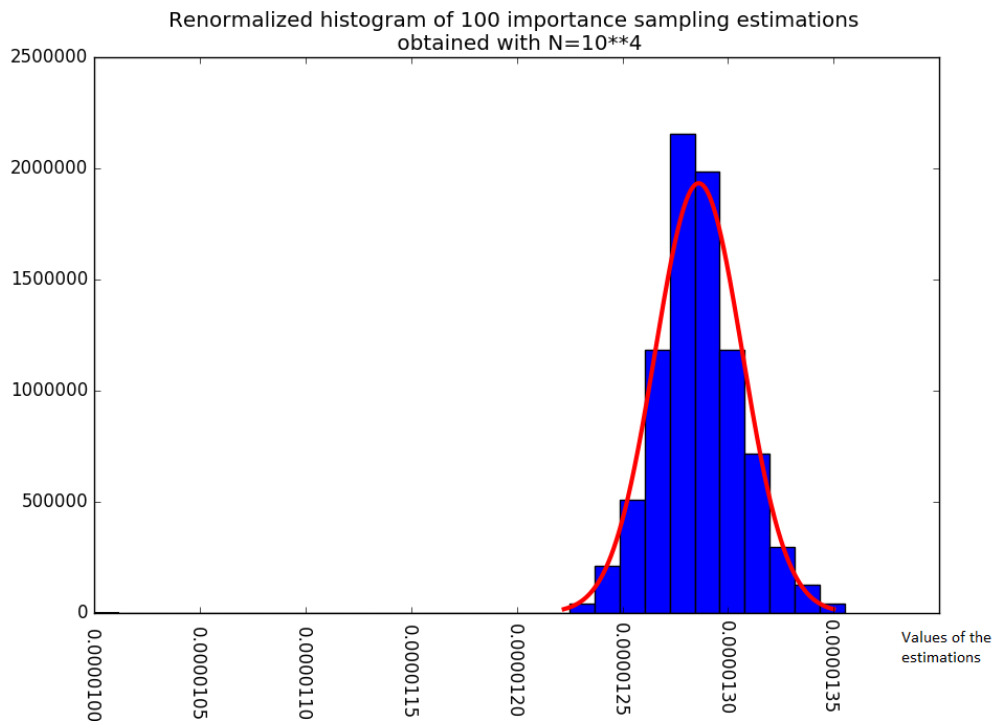


Figure 5.1 – Asymptotic normality of the IS estimator (for $N_{sim} = 10^4$)

the weights of the failing trajectories in the sample of one run of the IS method with $N_{sim} = 10^4$. The Figure 5.2 shows that the weights are close to the value p , suggesting that the importance density is close to the optimal density. The figure 5.3 is a zoom-in on the largest weight: It shows there is no degenerated preponderant weight such that $\frac{f(\mathbf{z}'_i)}{g_\alpha(\mathbf{z}'_i)} \gg p$, suggesting there is no sign of under-favored region of \mathcal{D} in g_α . Here we do not need to check the weight degeneracy in all parts of \mathcal{D} because, as we know the value of p , we can simply check the estimation are unbiased and normally distributed to ensure convergence is reached. Finally, in Figures 5.4 and 5.5, we present the graphs of two trajectories obtained respectively with the original process with density f and with the importance process selected by the CE method with density $g_{(\alpha_1, \alpha_2)}$.

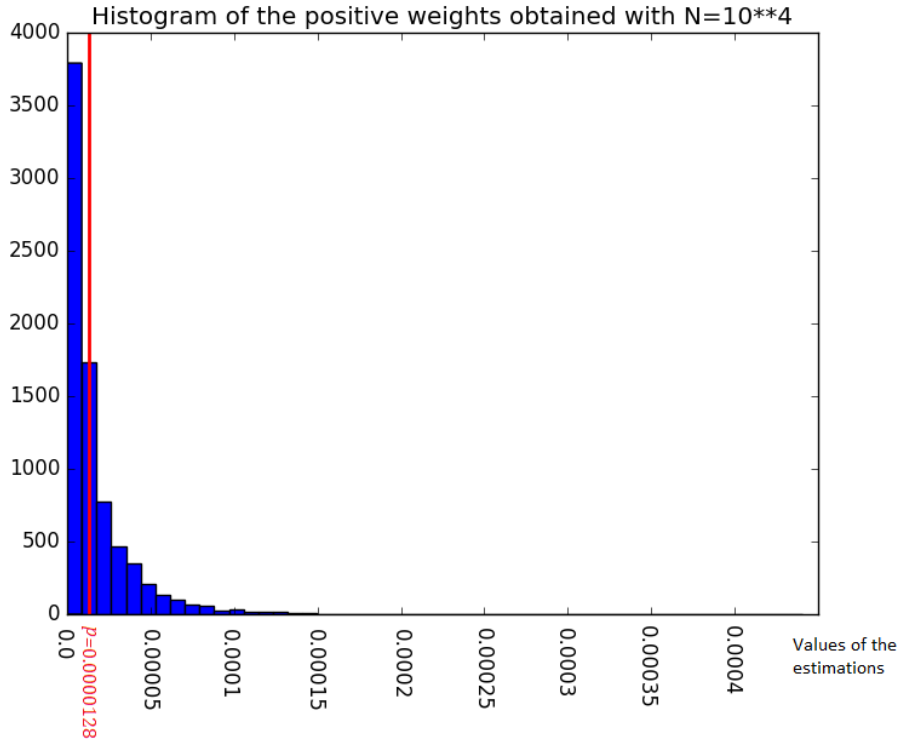


Figure 5.2 – Allocation of the weights of failing trajectories (for $N_{sim} = 10^4$)

5.2 Practical issues with power generation systems

When we want to apply importance sampling on power generation systems we pursue two main goals: Firstly, we want to specify a family of parametric densities g_α that are flexible enough to include a density close to the optimal density g^* , and secondly, we need to find a parameter α^* that yields a density g_{α^*} as close as possible to g^* . As we showed in the previous section, we can also transpose these goals in terms of the proximity function: instead of the density g^* , one can consider the optimal proximity function U^* , and instead of the densities g_α , one can consider the proximity functions U_α . The goals are then to specify a family of parametric functions U_α that are flexible enough to include a function close to an optimal proximity function U^* , and secondly, we need to find a parameter α^* that yields a function U_{α^*} as close as possible to a function proportional to U^* . In this section we presents some heuristic reasoning to build the family of functions U_α for any system. Despite our effort we did not have the time to develop a rigorous method to do so, but we believe that finding a rigorous way to specify the family of U_α is a important point of interest for future research, so we present here the ideas we had on the matter.

5.2.1 Specify a parametric family of proximity functions

We have seen in the introduction (subsections 2.2.5 and 2.3.2), that if we are not cautious enough we can design a parametric family of importance densities that is not

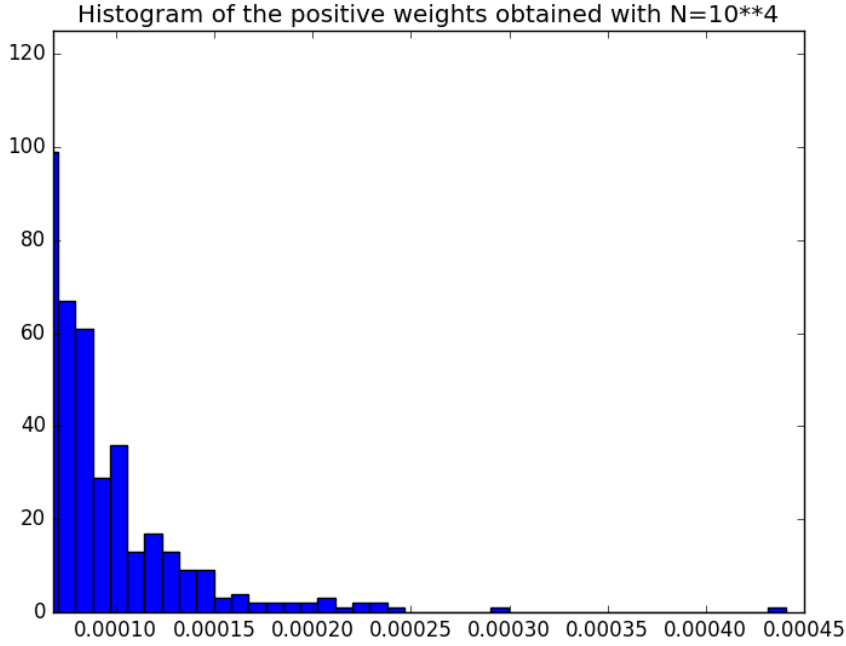


Figure 5.3 – Allocation of the largest weights in the sample (for $N_{sim} = 10^4$) (zoom-in of Figure 5.2)

flexible enough, and this can lead to misleading and biased estimates. Indeed, a density of importance should increase (in comparison to f) the density of the trajectories in \mathcal{D} homogeneously: ideally we want that

$$\forall \mathbf{z} \in \mathcal{D}, \text{ and } C \in [0, 1 - p], \quad g(\mathbf{z}) \geq \frac{f(\mathbf{z})}{C + p}, \quad (5.13)$$

with C as close as possible from 0 (See the property 5). But if we choose an importance density for which the increase of the density in comparison to f is inhomogeneous within \mathcal{D} , it can lead to an unnoticed slow convergence and to a non-conservative underestimation of p . (See section 2.2.5.)

Note that the set \mathcal{D} is difficult to apprehend, mainly because we have no obvious metric on the set of trajectories. Still we can already note that there are different types of failing trajectories, with very different skeletons, and so it seems unlikely to us that this set could be convex if we had such a metric. In order to control the increase of the density on all the trajectories of the set \mathcal{D} , we believe it is important to first identify all the parts of \mathcal{D} , and then, to ensure that the chosen importance density increases the probability of each of these parts proportionally.

This is why we have searched a way to identify all the different types of failing trajectories: We believe a good way to do so is to use the minimal groups of the systems.

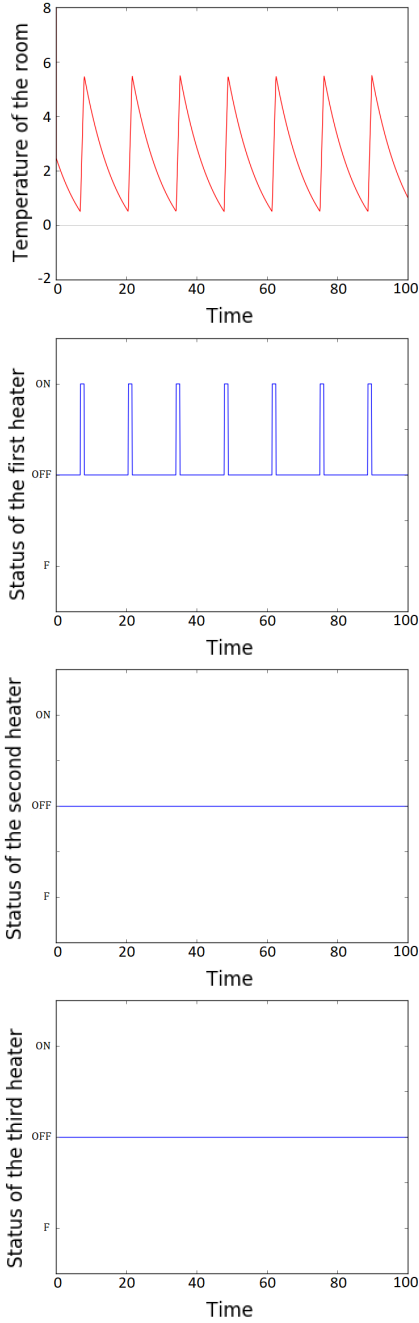


Figure 5.4 – A trajectory of the coordinates of the state of the system. This trajectory was generated with the original process with density f .

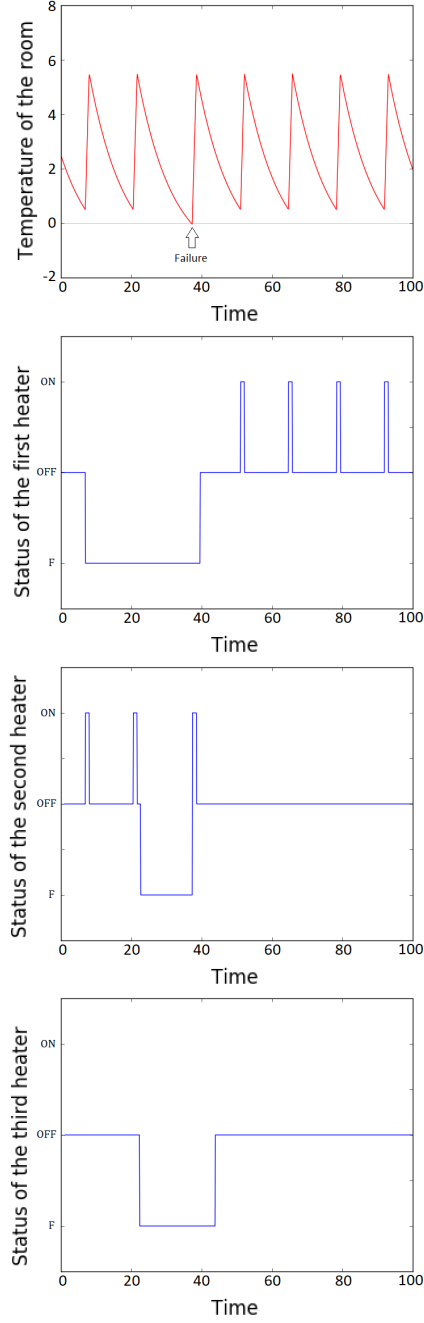


Figure 5.5 – A trajectory of the coordinates of the state of the system. This trajectory was generated with the importance process with density $g(\alpha_1, \alpha_2)$.

Minimal groups and minimal cut sets

How does the system moves from a safe state to a failing one (i.e a state in region D)? The system needs to go through successive failures of components that degrade the system to such a point that it becomes unable to run its main function. Then the system needs to stay in this degraded mode long enough so its state goes to D .

In order to classify the different types of failing trajectories we look for the causes of the failure. The first cause of the failure is the failure of a vital group components. The second cause of the system failure is the failure of a vital group components lasting long enough so that the system states goes to the failure region. The notion of vital group corresponds to the notion of Minimal group in a dynamical context, and it in a static context it rather corresponds to the minimal cut sets. A minimal group is a group of components having two characteristics 1) if all the components of the group stay failed the system will fail, 2) A permanent repair of one component within the group prevents the system from failing. A minimal cut sets is a group of components for which the system is failed when the components of the minimal cut sets are failed. The difference between minimal group and a minimal cut set is that, for a static model of the system, the system fails as soon as all the components of a minimal cut sets are failed, whereas, for a dynamic model of the systems, the systems fails only when the components of a minimal group have staid failed long enough so that the state of the system reaches the failure zone. Also note that a group of component is not necessarily a minimal groups at any time. Indeed the surrounding conditions (for instance the height of the overflow for a dam) can determine if a group of components is minimal or not: if the surrounding conditions change over time some group of components can be minimal only a portion of the observation time. For dynamic system we would rather consider the notion of minimal sequence. A minimal sequence is like a minimal group in which the failures of the components are chronologically ordered. But as the minimal sequences of a system are generally harder to find then its minimal groups, we choose to focus only on minimal groups.

To each failing trajectory we can assign one or more minimal groups that were failed at the moment of the system failure τ_D . For a minimal group indexed by i we denote by \mathcal{D}_i the set of trajectories within \mathcal{D} for which the i -th minimal group is failed at τ_D . If I_{minimal} is the set of the indices of the minimal groups, we have

$$\mathcal{D} = \bigcup_{i \in I_{\text{minimal}}} \mathcal{D}_i. \quad (5.14)$$

Note that the sets \mathcal{D}_i do not form a partition of \mathcal{D} as we may have $\mathcal{D}_i \cap \mathcal{D}_j \neq \emptyset$. Indeed, if for a trajectory \mathbf{z} the groups i and j are failed at τ_D , then $\mathbf{z} \in \mathcal{D}_i \cap \mathcal{D}_j$.

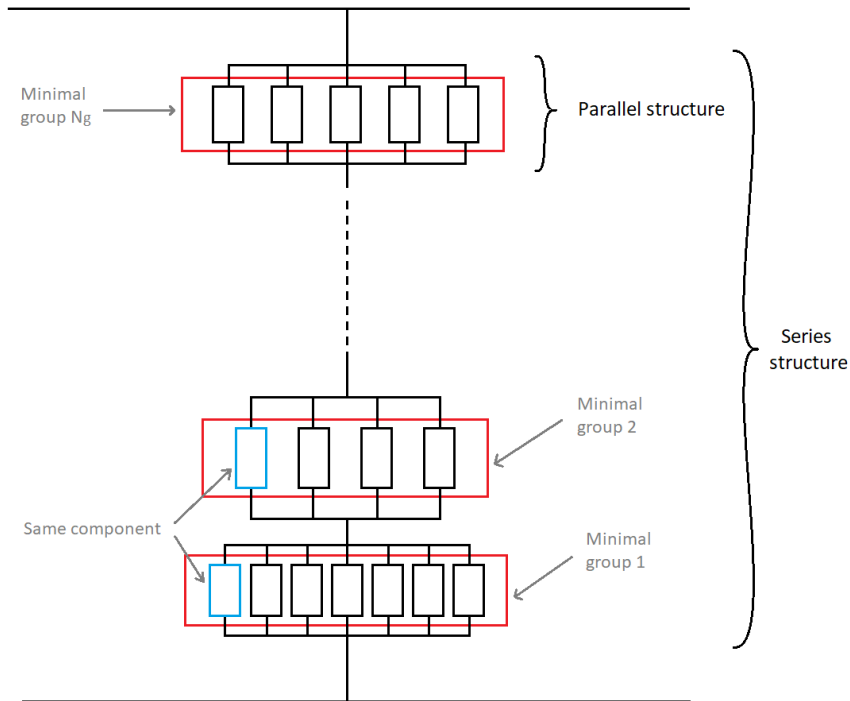


Figure 5.6 – Alternative reliability diagram associated to a system with N_g minimal groups

Identification of the minimal groups

In order to identify the minimal groups we suggest a heuristic method. We propose to make a static approximation of the system by considering that it fails as soon as the components maintaining its main functionalities are failed. We therefore neglect the time needed to go from this degraded state to a state in D . We are mixing up the notion of minimal group and of minimal cut set. This static approximation allows to use common tools of the field of reliability analysis of static systems [9]. Among those tools is the reliability diagram. This diagram indicates which components need to break in order to reach a failing state. It gives the modes $m \in \mathbb{M}$ such that there is a position $x \in \Omega_m$, with $(x, m) \in D$. This diagram is in fact a representation of the logic function that tells if the statuses of the components $M_t \in \mathbb{M}$ are degraded enough so that the system is going to fail if no repair occurs. For instance, in the example of the spent-fuel pool in section 1.3.5, we have deduced the logic function (1.111) from the diagram 1.21. The minimal groups of the system can be identified by fully developing the expression of the Logic function of the system, each term of the developed sum identifying a minimal group. Given the size of the industrial systems this diagram can be very helpful to determine the Logic function of the system.

If we allow for replicates of the box of a component in the diagram, we can associate an alternative diagram to the developed form of the logic function. This diagram is a series diagram in which each part corresponds to the components of a minimal group in parallel. In figure 5.6 we represent the alternative diagram of a system with N_g minimal groups. If

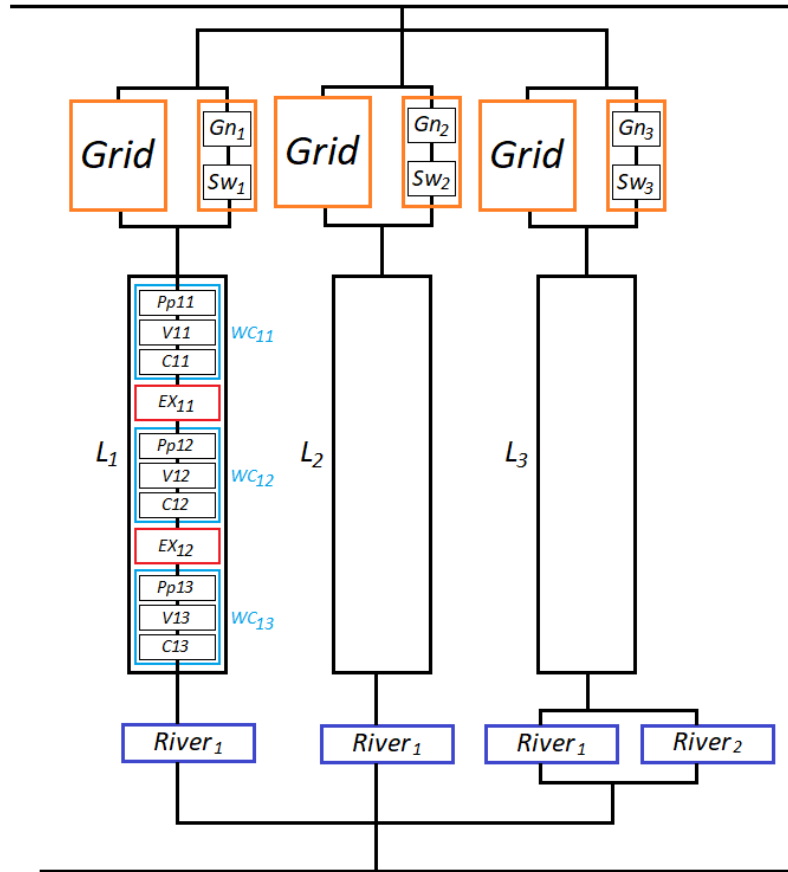


Figure 5.7 – Alternative series/parallel reliability diagram associated to the spent fuel pool system

we allow for replicates of the box of a component in the diagram, we can also represent the diagram as a diagram using only parallel and series structures. For instance, for the spent fuel pool one could transform the diagram 1.21 into the diagram presented in Figure 5.7. For big systems, these forms of reliability diagrams with replicates are probably easier to build, and it can still be used to identify the logic function and the minimal groups.

This approach being based on a static approximation of the system, it might be insufficient for complex systems, especially if some components reconfigure the structure of the system, based on dynamic events like a physical variable crossing a threshold. For such more complex systems it might be an option to include boxes in the diagram that represent the crossing of the thresholds, and to treat this event like the failure of a component, therefore adding Boolean associated to the event in the logic function of the system. Yet we did not have the time to investigate this idea.

A family of parametric proximity functions based on minimal groups

Having an importance density that can increase the density proportionally in all sets \mathcal{D}_i , seems a good strategy to get a well proportioned importance density. We present here two possible ways to specify a family of parametric importance densities that are

flexible enough so that, getting a homogeneous increase of the density can be achieved by choosing the right sets of parameters.

We start by presenting an option that we could not test as it requires to implement new functionalities in the PyCATSHOO toolbox. In this option, we assume that the probability associated to the trajectories with more than one minimal group failed at τ_D is negligible, so that for a state z before the failure:

$$U^*(\mathbf{z}_s, s) = \mathbb{E} \left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) | Z_s = z \right] \simeq \sum_{i \in I_{\text{minimal}}} \mathbb{E} \left[\mathbb{1}_{\mathcal{D}_i}(\mathbf{Z}_{t_f}) | Z_s = z \right]. \quad (5.15)$$

Then we approximate each term of the sum, $U_{\alpha,i}(\mathbf{z}_s, s)$ being the approximation of $E \left[\mathbb{1}_{\mathcal{D}_i}(\mathbf{Z}_{t_f}) | Z_s = z \right]$ so that

$$U_{\alpha}(\mathbf{z}_s, s) = \sum_{i \in I_{\text{minimal}}} U_{\alpha,i}(\mathbf{z}_s, s).$$

We let i_j be the index of the j -th component in the i -th minimal group, and let $N_{\text{min},i}$ be the number of components in the i -th minimal group. For $j \in \{1, \dots, N_c\}$, we denote by α_j the parameter associated to the contribution of the failure of the j -th component of the system to U_{α} . Let α_{N_c+1} be a shape parameter. For a state $z = (x, (m, 0))$ we denote by $z_{\text{min},i} = (x, (m', 0))$ the state identical to z except that the i -th minimal group is failed in m' : $\forall j \in \{1, \dots, N_{\text{min},i}\} m'^{i_j} = F$, and for other coordinates j , $m'^j = m^j$. We denote by $\lambda_{z_{\text{min},i}}^{\text{rep}}$ the sum of the repair rates of the broken components in state $z_{\text{min},i}$. We propose to take if $\tau_D > s$

$$U_{\alpha,i}(\mathbf{z}_s, s) = \exp \left[- \int_0^{t_{z_{\text{min},i}}^*} \lambda_{z_{\text{min},i}}^{\text{rep}}(u) du + \left(\sum_{j=1}^{N_{\text{min},i}} \alpha_{i_j} \mathbb{1}_{M_s^{i_j} \neq F} \right)^{\alpha_{N_c+1}} \right].$$

and if $\tau_D < s$, $U_{\alpha,i}(\mathbf{z}_s, s) = 1$. But this would make too many parameters for the optimization method of the CE method, so we propose to make groups of components according to their repair and jumping rates and failure on demand probabilities, and to set the parameters α_j of the components of the same group equal. This way we drastically reduce the number of parameters to optimize. The number of groups is left to the practitioner, keeping in mind that the more groups we take the more flexible is the parametric approximation but the less easy is the optimization of the parameters.

A family of parametric proximity functions based on series/parallel approximations of the system

As we have noticed earlier, we can also have an alternative diagram that is a series/parallel diagram. With this kind of static diagrams it is possible to express the probability of system failure using products and sums of the probabilities of failure of the

components. When some components form a system and are organized in series, one can sum the probabilities of failure to get a good approximation of the system probabilities of failure. Similarly when the components are in parallel, one can multiply the probability of failure to get a good approximation of the system probability of failure. Using this logic and if we consider that each component has a probability of failing $\exp(-\alpha_i)$, we get that if $\tau_D > s$:

$$U_{\alpha,i}(\mathbf{z}_s, s) = \exp \left[\sum_{j=1}^{N_{min,i}} \alpha_{i_j} \mathbb{1}_{M_s^{i_j} \neq F} \right].$$

and if $\tau_D < s$, $U_{\alpha,i}(\mathbf{z}_s, s) = 1$. Here again one can regroup the parameters of the components sharing similar characteristics, and set their parameters equal to reduce the number of parameters.

5.2.2 Results with the series/parallel parametric proximity function on the spent-fuel-pool system

We have tried to run the Cross-Entropy method with this family of parametric proximity functions, see the algorithm 2.1. As generating some failing trajectories directly with the original system (i.e. with $\alpha_0 = 0$) is too computationally demanding, we try to start the algorithm with different values of α_0 . Unfortunately the CE method has not identified the same minimum each time. We have tried to run the IS with each value of the parameters suggested by the CE method, but we have obtained very different and biased estimations of p . In this section we discuss the reasons of our method failure, and provide some ideas to improve the method.

Remember the CE consists in finding the parameter α^* such that

$$\alpha^* \in \operatorname{argmax}_{\alpha \in A_{param}} \mathbb{E}_f \left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \log(g_\alpha(\mathbf{Z}_{t_f})) \right].$$

In our case α^* can not be computed analytically because the derivative of $\alpha \rightarrow \mathbb{E}_f \left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \log(g_\alpha(\mathbf{Z}_{t_f})) \right]$ is unknown, so we rely on an approximation of it. $\mathbb{E}_f \left[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \log(g_\alpha(\mathbf{Z}_{t_f})) \right]$ is approximated by:

$$\frac{1}{N_{CE,0}} \sum_{i=1}^{N_{CE,0}} \mathbb{1}_{\mathcal{D}}(\mathbf{Z}_i) \log(g_\alpha(\mathbf{Z}_i)) \frac{f(\mathbf{Z}_i)}{g_{\alpha_0}(\mathbf{Z}_i)}, \quad \text{where } Y_i \sim g_{\alpha_0}. \quad (5.16)$$

And the first approximation of α^* , denoted by α_1 , can now be given by solving:

$$\frac{1}{N_{CE,0}} \sum_{i=1}^{N_{CE,0}} h(\mathbf{Z}_i) \nabla_\alpha \log(g_\alpha(\mathbf{Z}_i)) \frac{f(\mathbf{Z}_i)}{g_{\alpha_0}(\mathbf{Z}_i)} = 0, \quad \text{where } Y_i \sim g_{\alpha_0}. \quad (5.17)$$

Then we repeat the process, getting an $\alpha_2, \alpha_3, \dots$ and soon on until the value of α_n is

stabilized.

There are several explanations for the misleading results we obtained. The slow convergence observed can be related to our choice of the family $(U_\alpha)_{\alpha \in A_{param}}$ that may result in importance densities g_α that favor some areas of \mathcal{D} over others in the density. The slow convergence can also be related to an approximation error in equation 5.16, leading to a wrong selection of the parameter α . Furthermore, if we make an approximation error at the first step of the CE, we may bias the whole parameter optimization. For instance if the density g_{α_0} favors an area $\mathcal{D}_{privileged, \alpha_0}$ of \mathcal{D} over others, we may not have realizations \mathbf{Z}_i in $\mathcal{D} \setminus \mathcal{D}_{privileged, \alpha_0}$. Consequently, as all the \mathbf{Z}_i are in $\mathcal{D}_{privileged, \alpha_0}$, we could replace $\mathbb{1}_{\mathcal{D}}$ by $\mathbb{1}_{\mathcal{D}_{privileged, \alpha_0}}$ in the equation 5.16 without changing the approximation, the situation would be the same as if we were estimating $\mathbb{P}(\mathbf{Z}_{t_f} \in \mathcal{D}_{privileged, \alpha_0})$, and the optimization of α will yield a parameter for which g_α only privileges the region $\mathcal{D}_{privileged, \alpha_0}$. The issue is that we lack diagnostic tools to identify in which of these two situations we are.

To prevent such errors in the optimization we believe it would be good to start the optimization only after we get some realizations \mathbf{Z}_i in all the parts \mathcal{D}_j . The problem, is that it would require that the number of simulations $N_{CE, k}$ be very high, which we can not afford. Although we have not found a rigorous way to do it, we think it would be interesting to use a trajectory \mathbf{Z}_i in each part \mathcal{D}_j that is not generated but fully constructed by the practitioner from its skeleton. This way all the parts of \mathcal{D}_i will be represented in the sample, which could give a better estimations of $\mathbb{E}_f[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f}) \log(g_\alpha(\mathbf{Z}_{t_f}))]$. In order to check if the importance density homogeneously increase in \mathcal{D} , we could ensure that the likelihood-ratios for these trajectories in all the parts \mathcal{D}_j , have roughly the same values and that these values are lower than one.

Conclusion on the importance sampling for PDMP

Our work makes the use of importance sampling on PDMPs with boundaries possible, and it provides a convenient way to specify the law of the importance process. We have obtained significant variance reduction on a small system, yet more work is needed to determine an efficient importance process for systems of industrial size.

We have developed the theoretical foundations of importance sampling for PDMPs with boundaries. The challenge is to properly define the likelihood ratio involved in the weighting of the simulation outputs in importance sampling when the random variable of interest is a trajectory of PDMP with boundaries. We have exhibited the reference measure ζ_{t_f} on the space of PDMP trajectories, which allowed us to identify the closed form expression of the density of a PDMP trajectory, and to properly define the likelihood ratio. Thanks to the reference measure, we also have been able to identify the possible importance processes.

Then we have studied the characteristics of the optimal process with density $g^*(\mathbf{z}) = \frac{|h(\mathbf{z})|f(\mathbf{z})}{p}$. We have managed to show that the law of this optimal process is connected to a function U^* defined by:

$$U^* : \bigcup_{s \in [0, t_f]} \mathbf{E}_s \times \{s\} \longrightarrow \mathbb{R}$$

$$(\mathbf{z}_s, s) \longrightarrow \mathbb{E}[h(\mathbf{Z}_{t_f}) | \mathbf{Z}_s = \mathbf{z}_s]$$

and to the function $U^- = \int U^* d\mathcal{K}$. Indeed, we have shown that the density of the time between, for instance, the k -th jump and of the $k+1$ -th jump is such that, $\forall s \in [0, t_f)$, $\mathbf{z}_s \in \mathbf{E}_s$, $u \in [0, t_{z_s}^*)$:

$$g_{T'_k | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s + u)}{U^*(\mathbf{z}_s, s)} f_{T_k | Z_{S_k} = z_s}(u).$$

We have shown that the transition kernels and the jump rates associated to this optimal

process are such that, $\forall s \in [0, t_f)$, $\mathbf{z}_s \in \mathbf{E}_s$, $u \in [0, t_{z_s}^*)$:

$$K_{\mathbf{z}_s^-, s}^*(z) = \frac{U^*(\mathbf{z}_s, s)}{U^-(\mathbf{z}_s^-, s)} K_{z_s^-}(z_s),$$

$$\lambda_{\mathbf{z}_s, s}^*(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s + u)}{U^*(\Phi_{\mathbf{z}_s}(u), s + u)} \lambda_{z_s}(u).$$

Our approach through the function U^* can be applied to any sub-classes of PDMP, like, for instance, Markov chains [35], or continuous time Markov Chain, or queuing models. In the particular case of PDMP that is a continuous time Markov Chain, the definition of the function U^* is close to the forward committor function used in the transition path theory [42]. In the case of a general PDMP, a committor function would be a function $(z, s) \rightarrow \mathbb{E}[\mathbb{1}_{\mathcal{D}_A}(\mathbf{Z}) | Z_s = z]$ where \mathcal{D}_A is the set of trajectories that pass through D without passing through a set $A \subset E$ first. U^* is therefore a committor function for which $A = \emptyset$. It is also interesting to note that, in the Adaptive Multilevel Splitting algorithm, the asymptotic variance is minimized when using the committor function as the score function [5, 10], similarly, in the interacting particles system method [23], the function U^* also plays a role in the optimal potential function method which is a result proved in the part III and in [16]. Approaching the function U^* , allows to efficiently estimate rare event for importance sampling, but also for Adaptive Multilevel Splitting algorithm and the interacting particles system method. A method that allows to approximate this function would lead to significant improvement in the reliability assessment field.

As finding an efficient importance density is easier when we restrict the search to a family of parametric importance densities, we propose a way to specify the family of parametric importance densities. We have proposed to approximate the function U^* by parametric functions U_α , and to plug this approximation instead of U^* into the expressions of the optimal jump rates and of the transition kernels, which yields an importance process with some transition kernels and jump rates such that, $\forall s \in [0, t_f)$, $\mathbf{z}_s \in \mathbf{E}_s$, $u \in [0, t_{z_s}^*)$:

$$K'_{\mathbf{z}_s^-, s}(z_s) = \frac{U_\alpha(\mathbf{z}_s, s)}{U_\alpha^-(\mathbf{z}_s^-, s)} K_{z_s^-}(z_s),$$

$$\lambda'_{\mathbf{z}_s, s}(u) = \frac{U_\alpha^-(\Phi_{\mathbf{z}_s}(u), s + u)}{U_\alpha(\Phi_{\mathbf{z}_s}(u), s + u)} \lambda_{z_s}(u).$$

Passing through the expressions of the jump rates being beneficial because it allows to avoid computing renormalization constants during the generation of the trajectories of the importance process. We propose to then optimize the parameters with the Cross-entropy

method.

We have applied our importance sampling method to two systems: the heated room system and the spent-fuel-pool system. The method has worked very well on the heated room system, but has failed on the bigger system of the spent-fuel-pool. This shows our importance sampling method can work very well but it needs to be improved to be efficient on industrial systems.

We believe the failure on the industrial system is due to two main factors. The first reason being that the system is complex and has many ways to fail. Indeed the parametric function U_α must give an importance density that favors all the ways to fail, which is difficult to do. The second reason is that the cross-entropy is not well suited to find the optimal parameters when the failure region is not convex and is made of many parts.

PART III

A contribution to the IPS method

The second focus on this thesis is the interacting particle system (IPS) method, that was presented in section 2.4. The results of this part are also presented in the article [16]. Like importance sampling, the IPS method can provide a significant variance reduction, so we would like to use it for the reliability assessment of Power generation system too. Before considering the application of the method on PDMP, we shortly present a simple and useful theoretical result on the IPS method. We essentially choose to present this result in a separated part in order to emphasize that it is not necessarily related to a PDMP model, and that it can be used for a broader spectrum of models.

Indeed, in this part the model is a Markovian chain $(Z_k)_{k \leq n}$ with values in measurable sets (E_k, \mathcal{E}_k) for $k \leq n$. For instance, it could be the realization of a Gaussian process at successive times, or a simple random walk. The notations of this part are consistent with the notation of the section 2.4 in the introduction. In particular we consider the trajectories $\mathbf{Z}_k = (Z_0, Z_1, \dots, Z_k)$ with values in the measurable sets $(\mathbf{E}_k, \mathcal{E}_k)$ where $\mathbf{E}_k = E_0 \times E_1 \times \dots \times E_k$ and $\mathcal{E}_k = \mathcal{E}_0 \times \mathcal{E}_1 \times \dots \times \mathcal{E}_k$ for $k \leq n$. The Markovian chain $(Z_k)_{k \leq n}$ has a transition kernel v_k such that for $k > 0$ and for any bounded measurable function $t : E_k \rightarrow \mathbb{R}$

$$\mathbb{E}[t(Z_k)|Z_{k-1}] = \int_{E_k} t(z_k)v_k(dz_k|Z_{k-1}).$$

For any bounded measurable function $t : E_0 \times \dots \times E_n \rightarrow \mathbb{R}$ we have:

$$\mathbb{E}[t(Z_0, \dots, Z_n)] = \int_{E_n \times \dots \times E_0} t(z_0, \dots, z_n)v_n(dz_n|z_{n-1}) \dots v_1(dz_1|z_0)v_0(dz_0)$$

For $i < j$, consider two trajectories \mathbf{z}_i and \mathbf{z}_j : when it is necessary to differentiate the coordinates of these trajectories we write the coordinates $z_{i,k}$ for $k \leq i$ and $z_{j,k}$ for $k \leq j$ such that $\mathbf{z}_i = (z_{i,0}, z_{i,1}, \dots, z_{i,i})$ and $\mathbf{z}_j = (z_{j,0}, z_{j,1}, \dots, z_{j,j})$. We introduce the Markov Chain of the trajectories $(\mathbf{Z}_k)_{k \geq 0}$ with values in the measurable spaces $(\mathbf{E}_k, \mathcal{E}_k)$, and with the transition kernels \mathbf{V}_k such that:

$$\mathbf{V}_k(d\mathbf{z}_k|\mathbf{z}_{k-1}) = \delta_{\mathbf{z}_{k-1}}(d(z_{k,0}, \dots, z_{k,k-1}))v_k(dz_{k,k}|z_{k-1,k-1}).$$

The optimal potential functions

We have seen in section 2.4, that the IPS takes in input a sequence of potential functions $(G_k)_{k < n}$, and provides an estimator \hat{p}_G of $p = \mathbb{E}[h(\mathbf{Z}_n)]$ such that

$$\sqrt{N}(\hat{p}_G - p) \xrightarrow[N \rightarrow \infty]{d} \mathcal{N}(0, \sigma_{IPS,G}^2) \quad (7.1)$$

where, with the convention that $\prod_{i=0}^{-1} G_i(\mathbf{Z}_i) = \prod_{i=0}^{-1} G_i^{-1}(\mathbf{Z}_i) = 1$:

$$\sigma_{IPS,G}^2 = \sum_{k=0}^n \left\{ \mathbb{E} \left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_i) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n) | \mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] - p^2 \right\}. \quad (7.2)$$

So the amplitude of the variance reduction is associated to the choice of potential functions, yet there is not much information in the literature on how to choose these potentials functions $(G_k)_{k < n}$.

In this part we tackle the issue of the choice of the potential functions. Our contribution is to provide the expressions of the theoretical optimal potential functions that minimize the variance of the estimator of the IPS method. We hope these expressions will lead the practitioners to design more efficient potential functions, that are closer from the optimal ones.

Note that the result we present here, is not especially related to PDMP, and that it applies to any case of application of the IPS method.

7.1 The potentials used in the literature

As we said, the choice of the potential functions $(G_k)_{k < n}$ is paramount because it determines the variance of the IPS estimator, but so far, little information has been provided on the form of efficient potential functions. The standard approach is to find the best potential function within a set of parametric potential functions, and so the efficiency of the method strongly depends on the quality of the chosen parametric family.

For instance, in [23] the authors obtain their best variance reduction by choosing

$$G_k(\mathbf{Z}_k) = \frac{\exp[-\lambda R(Z_k)]}{\exp[-\lambda R(Z_{k-1})]}$$

where λ is a positive tuning parameter, and the quantity $R(z) = a - z$ roughly measures the proximity of z to the critical region that was $D = [a; +\infty)$.

In [53] the authors stress out that it seems better to take a time-dependent proximity function R_k instead of R , yielding:

$$G_k(\mathbf{Z}_k) = \frac{\exp[-\lambda R_k(Z_k)]}{\exp[-\lambda R_{k-1}(Z_{k-1})]},$$

where the quantities $R_k(z)$ are again measuring the proximity of z to D . Once the set of parametric potential functions is chosen, it is necessary to optimize the tuning parameters of the potentials. Different methods have been proposed. In [31], an empirical heuristic algorithm is provided; in [43] a meta model of the variance is minimized; in [23] the large deviation principle is used as a guide.

An other common option for the potential functions is the one done in splitting methods. Indeed the splitting method can also be seen as a version of the IPS method [12]. In this method one wants to assess the probability that a random variable Z belongs to a subset B_n . A succession of nested sets $E = B_0 \supseteq B_1 \supseteq B_2 \supseteq \dots \supseteq B_n$ is chosen by the practitioner or possibly chosen in an adaptive manner [11]. The small probability $\mathbb{P}(Z \in B_n)$ can then be decomposed into a product of conditional probabilities: $\mathbb{P}(Z \in B_n) = \prod_{i=1}^n \mathbb{P}(Z \in B_i | Z \in B_{i-1})$ and $\mathbb{P}(Z \in B_n) = \mathbb{E}[h(\mathbf{Z}_n)]$ by setting $h(\mathbf{Z}_n) = \mathbb{1}_{B_n}(Z_n)$. In this method the potential functions are chosen of the following form

$$G_k(\mathbf{Z}_k) = \mathbb{1}_{B_k}(Z_k).$$

One usually optimizes the variance reduction within this family of potential functions by optimizing the choice of the sets $(B_k)_{k \leq n}$.

7.2 The optimal potential

Note that if potential functions G_k and G'_k are such that $G_k = a.G'_k$ with $a > 0$, then they yield the same variance: $\sigma_{IPS,G}^2 = \sigma_{IPS,G'}^2$. Therefore the potential function is defined up to a multiplicative term.

Theorem 21. For $k \geq 1$, let G_k^* be defined by:

$$G_k^*(\mathbf{z}_k) \propto \begin{cases} \sqrt{\frac{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_{k+1}\right]^2|\mathbf{Z}_k=\mathbf{z}_k\right]}{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1}=\mathbf{z}_{k,0:k-1}\right]}} & \text{if } \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1}=\mathbf{z}_{k,0:k-1}\right] \neq 0 \\ 0 & \text{if } \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1}=\mathbf{z}_{k,0:k-1}\right] = 0 \end{cases} \quad (7.3)$$

and for $k = 0$,

$$G_0^*(\mathbf{z}_0) \propto \sqrt{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_1\right]^2|\mathbf{Z}_0=\mathbf{z}_0\right]}. \quad (7.4)$$

The potential functions minimizing $\sigma_{IPS,G}^2$ are the ones that are proportional to the G_k^* 's $\forall k \leq n$. The optimal variance of the IPS method with n steps is then

$$\begin{aligned} \sigma_{IPS,G^*}^2 &= \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_0\right]^2\right] - p^2 \\ &+ \sum_{k=1}^n \left\{ \mathbb{E}\left[\sqrt{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1}\right]} - p \right]^2 \right\}. \end{aligned} \quad (7.5)$$

Proof. As we lack mathematical tools to minimize $\sigma_{IPS,G}^2$ over the set of positive functions $(G_k)_{k \leq n}$, we had to guess the expressions (7.3) and (7.4) before providing the proof of the results. We begin this proof by presenting the heuristic reasoning that provided the expressions (7.3) and (7.4).

Assuming we already know the $k - 2$ first potential functions, we started by trying to find the $k - 1$ -th potential function G_{k-1} that minimizes the k -th term of the sum in (7.2). This is equivalent to minimize the quantity

$$\mathbb{E}\left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_i)\right] \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1}\right] \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s)\right] \quad (7.6)$$

over G_{k-1} . As the G_{k-1} are equivalent up to a multiplicative constant, we simplify the equation by choosing a multiplicative constant so that $\mathbb{E}\left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_i)\right] = 1$. Our minimizing problem then becomes the minimization of (7.6) under the constraint $\mathbb{E}\left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_i)\right] = 1$. In order to be able to use a Lagrangian minimization we temporarily assume that the distribution of \mathbf{Z}_{k-1} is discrete and that \mathbf{Z}_{k-1} takes its values in a finite or numerable

set \mathbf{E} . For $\mathbf{z} \in \mathbf{E}$, we denote $a_{\mathbf{z}} = \mathbb{P}(\mathbf{Z}_{k-1} = \mathbf{z})$ and $d_{\mathbf{z}} = \mathbb{E}[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2|\mathbf{Z}_{k-1} = \mathbf{z}]$ and $g_{\mathbf{z}} = \prod_{i=0}^{k-2} G_i(\mathbf{z}_i)G_{k-1}(\mathbf{z})$. Our minimization problem becomes the minimization of

$$\mathcal{L} = \left(\sum_{\mathbf{z} \in \mathbf{E}} \frac{a_{\mathbf{z}} d_{\mathbf{z}}}{g_{\mathbf{z}}} \right) - \lambda \left(1 - \sum_{\mathbf{z} \in \mathbf{E}} a_{\mathbf{z}} g_{\mathbf{z}} \right) \quad (7.7)$$

Finding the minimum of this Lagrangian we get that $g_{\mathbf{z}} = \frac{\sqrt{d_{\mathbf{z}}}}{\sum_{\mathbf{z}' \in \mathbf{E}} a_{\mathbf{z}'} \sqrt{d_{\mathbf{z}'}}}$. Now relaxing the constraint of the multiplicative constant, we get that

$$\prod_{i=0}^{k-1} G_i(\mathbf{z}_{k-1,0:i}) \propto \sqrt{\mathbb{E}[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2|\mathbf{Z}_{k-1} = \mathbf{z}_{k-1}]},$$

which gives the desired expressions. After these heuristic arguments we can now rigorously check that these expressions, obtained by minimizing each term of the sum in (7.2) one by one, also minimize the whole sum for any distribution of the \mathbf{Z}_{k-1} 's.

The proof now consists in showing that, for any set of potential functions $(G_s)_{s < n}$, we have $\sigma_{IPS,G}^2 \geq \sigma_{IPS,G^*}^2$. This is done by bounding from below each term of the sum in (7.2). We start by decomposing a product of potential functions as follows:

$$\forall k \in \{1, \dots, n\}, \quad \prod_{s=0}^{k-1} G_s(\mathbf{z}_{k-1,0:s}) = \epsilon_{k-1}(\mathbf{z}_{k-1}) \prod_{s=0}^{k-1} G_s^*(\mathbf{z}_{k-1,0:s}) + \bar{\epsilon}_{k-1}(\mathbf{z}_{k-1}) \quad (7.8)$$

where when $\mathbf{z}_{k-1} \in \text{supp} \prod_{s=0}^{k-1} G_s^*$,

$$\epsilon_{k-1}(\mathbf{z}_{k-1}) = \frac{\prod_{s=0}^{k-1} G_s(\mathbf{z}_{k-1,0:s})}{\prod_{s=0}^{k-1} G_s^*(\mathbf{z}_{k-1,0:s})}, \quad \text{and } \bar{\epsilon}_{k-1}(\mathbf{z}_{k-1}) = 0$$

and when $\mathbf{z}_{k-1} \notin \text{supp} \prod_{s=0}^{k-1} G_s^*$,

$$\epsilon_{k-1}(\mathbf{z}_{k-1}) = 0, \quad \text{and } \bar{\epsilon}_{k-1}(\mathbf{z}_{k-1}) = \prod_{s=0}^{k-1} G_s(\mathbf{z}_{k-1,0:s}).$$

Using (7.8) we get that

$$\begin{aligned} & \mathbb{E} \left[\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] \\ &= \mathbb{E} \left[\epsilon_{k-1}(\mathbf{Z}_{k-1}) \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right] \mathbb{E} \left[\mathbb{E}[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2|\mathbf{Z}_{k-1}] \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] \\ &+ \mathbb{E} \left[\bar{\epsilon}_{k-1}(\mathbf{Z}_{k-1}) \right] \mathbb{E} \left[\mathbb{E}[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2|\mathbf{Z}_{k-1}] \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right]. \end{aligned}$$

so

$$\begin{aligned} & \mathbb{E} \left[\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] \\ & \geq \mathbb{E} \left[\epsilon_{k-1}(\mathbf{Z}_{k-1}) \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right] \mathbb{E} \left[\frac{\mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} \right]}{\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s)} \right] + 0 \end{aligned} \quad (7.9)$$

For $\mathbf{z}_{k-1} \in \text{supp} \prod_{s=0}^{k-1} G_s^*$ we have:

$$\prod_{s=0}^{k-1} G_s^*(\mathbf{z}_{k-1,0:s}) \propto \sqrt{\mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} = \mathbf{z}_{k-1} \right]}$$

So $\text{supp} \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} = \mathbf{z}_{k-1} \right] = \text{supp} \prod_{s=0}^{k-1} G_s^*$ and we get

$$\begin{aligned} \mathbb{E} \left[\frac{\mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} \right]}{\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s)} \right] &= \mathbb{E} \left[\frac{\mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} \right]}{\epsilon_{k-1}(\mathbf{Z}_{k-1}) \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s)} \right] \\ &= \mathbb{E} \left[\frac{1}{\epsilon_{k-1}(\mathbf{Z}_{k-1})} \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right]. \end{aligned} \quad (7.10)$$

Combining (7.10) with inequality (7.9) we get that

$$\begin{aligned} & \mathbb{E} \left[\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] \\ & \geq \mathbb{E} \left[\epsilon_{k-1}(\mathbf{Z}_{k-1}) \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right] \mathbb{E} \left[\frac{1}{\epsilon_{k-1}(\mathbf{Z}_{k-1})} \prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right] \end{aligned} \quad (7.11)$$

and using the Cauchy-Schwarz inequality on the right term, we get that

$$\begin{aligned} & \mathbb{E} \left[\prod_{s=0}^{k-1} G_s(\mathbf{Z}_s) \right] \mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_s) \right] \\ & \geq \mathbb{E} \left[\prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right]^2 = \mathbb{E} \left[\prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s) \right] \mathbb{E} \left[\frac{\mathbb{E} \left[\mathbb{E}[h(\mathbf{Z}_n)|\mathbf{Z}_k]^2 | \mathbf{Z}_{k-1} \right]}{\prod_{s=0}^{k-1} G_s^*(\mathbf{Z}_s)} \right]. \end{aligned} \quad (7.12)$$

By summing the inequalities (7.12) for each k , we easily see that

$$\sigma_{IPS,G}^2 \geq \sigma_{IPS,G^*}^2,$$

which completes the proof of the theorem. \square

Remark that the optimal potential may be not positive everywhere, unfortunately, so it violates the hypothesis under which the TCL was proven in [21]. We claim that this question of the positiveness of the potential has not much interest in practice. Assume we take potential functions $(G_k)_{k < n}$ such that in the equation (7.9) we have $\epsilon_k(\mathbf{Z}_k) = 1$ and

$\bar{\varepsilon}_k(\mathbf{Z}_k) = \varepsilon > 0$ with ε very small. Choosing ε small enough, we can get $(G_k)_{k < n}$ as close as we want from $(G_k^*)_{k < n}$. With such potential functions $(G_k)_{k < n}$ and ε small enough, it is very likely that we would obtain the same samples in the algorithm as if we had taken the potentials $(G_k^*)_{k < n}$, and so we would have the same estimation. Moreover, with such potential functions we would also have a TCL with a variance very close to σ_{IPS, G^*}^2 . (According to (7.9) by choosing ε as small as we want, we get a variance as close as we want from σ_{IPS, G^*}^2 .) In practice, positive potential functions $(G_k)_{k < n}$ with ε close to zero give the same results as the potentials $(G_k^*)_{k < n}$.

An other important remark is that the optimal potentials $(G_k^*)_{k < n}$ favor the particles with large increments of the quantity $\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n) \mid \mathbf{Z}_{k+1}\right]^2 \mid \mathbf{Z}_k = \mathbf{z}_k\right]$ rather than particles with a large value for this quantity.

7.3 A short simulation study for the comparison of potentials

In this section we confirm empirically the results of the previous section by carrying out a short simulation study. Here, we apply the IPS method on a toy system for which we have explicit formulas. The system under consideration is the Gaussian random walk $Z_{k+1} = Z_k + \varepsilon_{k+1}$, $Z_0 = 0$, where the $(\varepsilon_k)_{k \in \{1, \dots, n\}}$ are i.i.d. Gaussian random variables with mean zero and variance one. We explore two situations, one where the quantity to estimate, the optimal potential and the variance of the estimator can be calculated explicitly, and one where these quantities can be approximated by a large deviation inequality.

7.3.1 First example

In the first situation, taking $b, a > 0$ and $n \in \mathbb{N} \setminus \{0\}$, the goal is to compute the expectation $p = \mathbb{E}[h(\mathbf{Z}_n)]$ when $h(\mathbf{z}_n) = \exp[b(z_n - a)]$. As Z_n is a centered Gaussian of variance n , a simple calculation gives that for $k < n$:

$$\mathbb{E}[h(\mathbf{Z}_n) \mid Z_k = z] = \exp\left((n - k) \frac{b^2}{2} + b(z - a)\right). \quad (7.13)$$

Consequently we have that:

$$p = \exp\left(\frac{n}{2} b^2 - ab\right), \quad (7.14)$$

and that, for $k \geq 1$:

$$\sqrt{\frac{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_{k+1}\right]^2|\mathbf{Z}_k = \mathbf{z}_k\right]}{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1} = \mathbf{z}_{k,0:k-1}\right]}} = \exp\left[-\frac{b^2}{2} + b(z_{k,k} - z_{k,k-1})\right] \quad (7.15)$$

$$\text{with } \mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_1\right]^2|\mathbf{Z}_0 = \mathbf{z}_0\right] = \exp\left[\frac{(n+1)b^2}{2} + b(z_0 - a)\right], \quad (7.16)$$

or equivalently that:

$$G_k^*(\mathbf{z}_k) \propto \exp\left[b(z_{k,k} - z_{k,k-1})\right] \quad (7.17)$$

$$\text{with } G_0^*(\mathbf{z}_0) \propto \exp\left[bz_0\right]. \quad (7.18)$$

We can observe here that the optimal potential favors particles with a big increment of the value z_k rather than particles with a large value z_k , with confirms what was empirically observed in [23]. Using the equations (7.17) and (7.18) in (7.2), it can easily be shown that the variance of the IPS estimator with these optimal potential functions is:

$$\begin{aligned} \sigma_{IPS,G^*}^2 &= n \left(\exp(b^2) - 1 \right) \exp\left[nb^2 - 2ab \right] \\ &= n \left(\exp(b^2) - 1 \right) p^2. \end{aligned} \quad (7.19)$$

It is notable that we obtain an asymptotically optimal variance [28], i.e. a variance proportional to p^2 .

In order to confirm these theoretical results, we have carried out a simulation study. We have run the method 200 times with $N = 10^5$, $n = 10$ and different values of a and b , and for each of these values we have computed the mean of the estimation and the empirical variance of the estimation. The results are displayed in table 7.1, where we compare the theoretical value of p to the empirical mean of the 200 estimations, and σ_{IPS,G^*}^2 to the empirical variance. As the theoretical values are close to the empirical ones, this confirms that the method is unbiased and that the variance given by the equation (7.2) is the right one. We also compare the empirical variance to the variance of the Monte-Carlo estimator, showing that, on this example, the IPS method provides a significant variance reduction with the optimal potential, as the variance is reduced by at least a factor 10^4 on the considered cases.

a	b	p	σ_{IPS,G^*}^2	σ_{MC}^2	$mean(\hat{p})$	$\hat{\sigma}_{IPS,G^*}^2$
40	$\sqrt{\log(1 + \frac{1}{n})}$	6.98×10^{-6}	6.87×10^{-16}	6.98×10^{-6}	6.98×10^{-6}	6.72×10^{-16}
40	$\sqrt{\log(1 + \frac{2}{n})}$	9.51×10^{-8}	1.81×10^{-14}	9.51×10^{-8}	9.51×10^{-8}	1.76×10^{-14}
40	$\sqrt{\log(1 + \frac{3}{n})}$	4.69×10^{-9}	6.61×10^{-17}	4.69×10^{-9}	4.69×10^{-9}	7.37×10^{-17}
40	$\sqrt{\log(1 + \frac{4}{n})}$	4.50×10^{-10}	8.13×10^{-19}	4.50×10^{-10}	4.50×10^{-10}	7.39×10^{-19}
35	$\sqrt{\log(1 + \frac{1}{n})}$	3.27×10^{-5}	1.09×10^{-9}	3.27×10^{-5}	3.27×10^{-5}	9.69×10^{-10}
35	$\sqrt{\log(1 + \frac{2}{n})}$	8.04×10^{-7}	1.29×10^{-12}	8.04×10^{-7}	8.04×10^{-7}	1.35×10^{-12}
35	$\sqrt{\log(1 + \frac{3}{n})}$	6.07×10^{-8}	1.11×10^{-14}	6.07×10^{-8}	6.07×10^{-8}	1.06×10^{-14}
35	$\sqrt{\log(1 + \frac{4}{n})}$	8.19×10^{-9}	2.69×10^{-16}	8.19×10^{-9}	8.19×10^{-9}	2.85×10^{-16}

Table 7.1 – Theoretical and empirical comparisons (example 1)
 $N = 2 * 10^5, n = 10$

7.3.2 Second example

In the second situation, the goal is to compute the probability that Z_n exceeds a large positive value a . Therefore we take $h(\mathbf{z}_n) = \mathbb{1}_{[a;+\infty)}(z_n)$ so that

$$p = \mathbb{P}(Z_n \geq a). \quad (7.20)$$

In that case one can not compute $\mathbb{E}[h(Z_n)|Z_k = z]$ but the Chernov-Bernstein's inequality gives the following sharp exponential bound:

$$\mathbb{E}[h(\mathbf{Z}_n)|Z_k = z] \leq \exp \left[-\frac{(a - z)^2}{2(n - k)} \right], \quad (7.21)$$

from which we can deduce that:

$$\prod_{i=0}^{k-1} G_i^*(\mathbf{z}_{k-1,0:i}) = \sqrt{\mathbb{E} \left[\mathbb{E} \left[h(\mathbf{Z}_n) | \mathbf{Z}_k \right]^2 | \mathbf{Z}_{k-1} = \mathbf{z}_{k-1} \right]} \leq C_1 \exp \left[C_2 - \frac{(z_{k-1} - a)^2}{2(n - k + 2)} \right], \quad (7.22)$$

where C_1 and C_2 are some constants independent of \mathbf{z}_{k-1} . One can therefore try to set the product of the potentials equal to this upper bound, which yields:

$$\text{for } k \geq 1, \quad G_k(\mathbf{z}_k) \propto \exp \left[-\frac{(z_k - a)^2}{2(n - k + 1)} + \frac{(z_{k-1} - a)^2}{2(n - k + 2)} \right] \quad (7.23)$$

$$\text{and} \quad G_0(\mathbf{z}_0) \propto \exp \left[-\frac{(z_0 - a)^2}{2(n - 1)} \right]. \quad (7.24)$$

Similarly as for the first example, we have carried out a simulation study. We have run the method 200 times with $N = 10^5, n = 10$ and different values of a , and for each of

a	p	σ_{IPS,G^*}^2	σ_{MC}^2	$mean(\hat{p})$	$\hat{\sigma}_{IPS,G}^2$
$4\sqrt{n}$	$3.17 * 10^{-5}$?	$3.17 * 10^{-5}$	$3.18 * 10^{-5}$	$8.14 * 10^{-8}$
$5\sqrt{n}$	$2.87 * 10^{-7}$?	$2.87 * 10^{-7}$	$2.86 * 10^{-7}$	$1.88 * 10^{-11}$
$6\sqrt{n}$	$9.87 * 10^{-10}$?	$9.87 * 10^{-10}$	$9.67 * 10^{-10}$	$6.63 * 10^{-16}$
$7\sqrt{n}$	$1.28 * 10^{-12}$?	$1.28 * 10^{-12}$	$1.29 * 10^{-12}$	$4.47 * 10^{-21}$

Table 7.2 – Theoretical and empirical comparisons (example 2)
results obtained with $N = 10^5$ and $n = 10$

these values we have computed the empirical mean of the estimation and the empirical variance of the estimation. We compare these estimations with the actual values of p and the variance of the Monte-Carlo method σ_{MC}^2 , showing that the potentials built with the Chernov-Bernstein large deviation inequality and our formula yield a significant variance reduction. Indeed the variance reduction compared to the Monte-Carlo method is at least by a factor 8500, and at best by a factor $2.4 * 10^6$.

We have also compared the efficiency of different potentials. We run the method with 1) the potential used on a Gaussian random walk in [31]: $G_k(\mathbf{z}_k) = \exp[\alpha(z_k - z_{k-1})]$ where the parameter was optimized to $\alpha = 1.1$, 2) the potential built with the Chernov-Bernstein large deviation inequality, and 3) with the optimal potential that we computed thanks to quadrature methods. The parameter α was chosen after several tries of values between 0.5 and 1.5 with a step 0.1. The value selected was the one yielding the best results in term of variance, and it was $\alpha = 1.1$. The results are displayed in table 7.3, and show that indeed the potential functions $(G_k^*)_{k < n}$, where

$$G_k^*(\mathbf{z}_k) = \sqrt{\frac{\int_{\mathbb{R}} \left(\int_a^\infty \exp \left[-\frac{(z'_n - z'_{k+1})^2}{2(n-k+1)} \right] dz'_n \right)^2 \exp \left[-\frac{(z'_{k+1} - z_k)^2}{2} \right] dz'_{k+1}}{\int_{\mathbb{R}} \left(\int_a^\infty \exp \left[-\frac{(z'_n - z'_k)^2}{2(n-k)} \right] dz'_n \right)^2 \exp \left[-\frac{(z'_k - z_{k-1})^2}{2} \right] dz'_k}},$$

yield the best variance.

$G_k(\mathbf{z}_k)$	$mean(\hat{p})$	$\hat{\sigma}_{IPS,G}^2$
$\exp[\alpha(z_k - z_{k-1})]$	$1.04 * 10^{-6}$	$2.44 * 10^{-10}$
$\exp \left[-\frac{(z_k - a)^2}{2(n-k+1)} + \frac{(z_{k-1} - a)^2}{2(n-k+2)} \right]$	$1.03 * 10^{-6}$	$1.78 * 10^{-10}$
$G_k^*(\mathbf{z}_k)$	$1.04 * 10^{-6}$	$1.62 * 10^{-10}$

Table 7.3 – Comparisons of the efficiency of potentials (example 2)
results obtained for $p = 1.05 * 10^{-6}$ $N = 2000$, $n = 10$, $a = 15$, $\alpha = 1.1$

Conclusions and implications

In this part, we have given closed form expressions of the optimal potential functions for the IPS method with multinomial resampling, and for its minimal variance. The existence of optimal potential functions proves that the possible variance reduction of an IPS method is lower-bounded. The expressions have been validated analytically and have been empirically confirmed in toy examples.

Furthermore the results found in the literature seem consistent with our findings. Indeed, in [23] the authors made the observation that it seemed better to build a potential which is defined in terms of the increments of an energy function, this observation is confirmed as the optimal potential is the multiplicative increment of the quantity $\sqrt{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k\right]^2|\mathbf{Z}_{k-1} = \mathbf{z}_{k-1}\right]}$ which is then the optimal energy function. Also, the fact that in [53] the authors find better results with time-dependent potentials is explained by the fact that the expression of the optimal potential shows a dependency on k . Finally, as splitting methods can be viewed as a version of the IPS-method with indicator potential functions, our results show that the selections of splitting algorithms are not optimal, and could be improved by using information on the expectations $\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k = \mathbf{z}\right]$.

The optimal potential functions may be hard to find in practice. Indeed, the expectations $\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k = \mathbf{z}\right]$ play a big role in the expression of the optimal potentials, but if we are trying to assess $p = \mathbb{E}\left[h(\mathbf{Z}_n)\right]$, we typically lack information about the expectations $\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k = \mathbf{z}\right]$. If no information on the expectation $\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k = \mathbf{z}\right]$ is available, it might be preferable to use more naive variance reduction method, where no input functions are needed. In such context, the Weighted Ensemble (WE) method [1, 2] seems to be a good candidate, as it does not take in input potential functions but only a partition of the state space. Conversely if the practitioner has information about the expectation $\mathbb{E}\left[h(\mathbf{Z}_n)|\mathbf{Z}_k = \mathbf{z}\right]$, this information could be used to derive very efficient potentials.

The knowledge of these expectations is therefore crucial for a well optimized use of the IPS method, but it is interesting to remark that the same knowledge seems to be crucial for a well optimized importance sampling method. Indeed in part II we showed that, when it is used on a piecewise deterministic process, the optimal density of importance sampling depends on $\mathbb{E}\left[|h(\mathbf{Z}_{t_f})||Z_{[0,s]} = z_{[0,s]}\right]$ where $Z_{[0,s]}$ is the trajectory of the PDMP on an interval $[0, s]$. This confirms the well known fact [7] that, with a good knowledge of the dynamic of the process $(\mathbf{Z}_t)_{t \geq 0}$, the importance sampling method is preferable to the IPS.

PART IV

The interacting particle method for PDMPs

The inefficiency of IPS on concentrated PDMP

In the previous part, we have seen that a well tuned IPS methods can provide a significant variance reduction. We now want to adapt and apply this method to the PDMP case for the purpose of the reliability assessment of power generation systems. The work presented in this part led to the publication of [14] in the review *Chaos: An Interdisciplinary Journal of Nonlinear Science*.

Notations

We first start by defining the notations required to use the IPS on PDMPs. We mainly redefine here the notations of the section on the IPS 2.4, so that they coincide with the notations we used for the PDMPs in the chapter 1 of the introduction, and with the notations of part II on importance sampling.

$(Z_s)_{s>0}$ denotes a PDMP, as defined in the chapter 1 of the introduction: $Z_s = (X_s, M_s)$ represent the state of an industrial system at time s . X_s being the vector of the physical variable, and M_s being the vector of the statuses of the components within the system. Z_t is a random variable with value in the measurable space (E, \mathcal{E}) . The state space is denoted by E , the flow on the states is denoted by Φ , the jump rate is denoted by λ , and the jump transition kernel is denoted by \mathcal{K} . $D \subset E$ is the failure region of the system, τ_D is the first hitting time of D . Θ is the application that transform a trajectory into its skeleton.

For any time t , we denote by $\mathbf{Z}_t = (Z_s)_{s \in [0, t]}$ the trajectory of the states up to time t . \mathbf{E}_t is the set of the trajectories defined on the interval $[0, t]$ that satisfy equation (1.47) page 16, and \mathcal{E}_t a σ -algebra on \mathbf{E}_t such that $\mathcal{E}_t = \theta^{-1}(\mathcal{S}_t)$, \mathcal{S}_t being defined in page 74. \mathcal{D} is the set of the trajectories including a system failure and defined on the interval $[0, t_f]$, it corresponds the set of trajectories such that $\tau_D \in (0, t_f]$. We are interested in assessing

$$p = \mathbb{P}(\tau_D \leq t_f) = \mathbb{P}(\mathbf{Z}_{t_f} \in \mathcal{D}) = \mathbb{E}[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f})],$$

but we present the method in the more general case. If h is a measurable function on

$(\mathbf{E}_{\tau_n}, \mathcal{E}_{\tau_n})$, we present the method to assess the quantity:

$$p = \mathbb{E} \left[h(\mathbf{Z}_{t_f}) \right].$$

9.1 IPS with PDMP

We consider $n + 1$ times $(\tau_k)_{0 \leq k \leq n}$ such that $0 = \tau_0 < \tau_1 < \dots < \tau_n = t_f$, and we decompose the interval $[0, t_f]$ in a series of sub intervals $(\tau_k, \tau_{k+1}]$ where $k \in \{0, \dots, n-1\}$. For each $k \leq n$, \mathbf{Z}_{τ_k} is a trajectory on $[0, \tau_k]$, and for $k < n$ $\mathbf{Z}_{(\tau_k, \tau_{k+1}]}$ denote the piece of the trajectory on the interval $(\tau_k, \tau_{k+1}]$. Remember that $\mathbf{E}_{(\tau_k, \tau_{k+1}]}$ is the set of trajectories defined on the interval $(\tau_k, \tau_{k+1}]$ and following the flow Φ . We defined its sigma algebra by $\mathcal{E}_{(\tau_k, \tau_{k+1}]} = \Theta^{-1}(\mathcal{S}_{(\tau_k, \tau_{k+1}]})$ with $\mathcal{S}_{(\tau_k, \tau_{k+1}]}$ defined in page 73. We denote by ν_0 the distribution of Z_0 or \mathbf{Z}_{τ_0} , and for each k , $1 \leq k \leq n$ we denote by v_k the transition kernel of the $\mathbf{Z}_{(\tau_k, \tau_{k+1}]}$ such that for any bounded measurable function $h : (\mathbf{E}_{(\tau_k, \tau_{k+1}]}, \mathcal{E}_{(\tau_k, \tau_{k+1}]}) \rightarrow (\mathbb{R}, \mathcal{B}(\mathbb{R}))$,

$$\mathbb{E}[h(\mathbf{Z}_{(\tau_k, \tau_{k+1}]}) | \mathbf{Z}_{(\tau_{k-1}, \tau_k]} = \mathbf{z}_{(\tau_{k-1}, \tau_k)}] = \int_{\mathbf{E}_{(\tau_k, \tau_{k+1}]}} h(\mathbf{z}_{(\tau_k, \tau_{k+1}]}) v_k(d\mathbf{z}_{(\tau_k, \tau_{k+1}]}) | \mathbf{z}_{(\tau_{k-1}, \tau_k)}.$$

For any bounded measurable function $h : \mathbf{E}_{\tau_0} \times \dots \times \mathbf{E}_{\tau_n} \rightarrow \mathbb{R}$ we have:

$$\mathbb{E}[h(\mathbf{Z}_{\tau_n})] = \int_{\mathbf{E}_{\tau_0} \times \mathbf{E}_{(\tau_0, \tau_1]} \times \dots \times \mathbf{E}_{(\tau_{n-1}, \tau_n]}} h(\mathbf{z}_{\tau_n}) v_n(d\mathbf{z}_{(\tau_{n-1}, \tau_n]} | \mathbf{z}_{(\tau_{n-2}, \tau_{n-1}]}) \dots v_1(d\mathbf{z}_{(\tau_0, \tau_1]} | z_0) \nu_0(dz_0)$$

Then we let \mathbf{V}_k be the Markovian kernel that extends \mathbf{Z}_{τ_k} into $\mathbf{Z}_{\tau_{k+1}}$, such that

$$\mathbf{V}_{k+1}(d\tilde{\mathbf{z}}_{\tau_{k+1}} | \mathbf{z}_{\tau_k}) = v_{k+1}(d\tilde{\mathbf{z}}_{(\tau_k, \tau_{k+1}]} | \mathbf{z}_{(\tau_{k-1}, \tau_k)}) \delta_{\mathbf{z}_{\tau_k}}(d\tilde{\mathbf{z}}_{\tau_k} \cdot)$$

For any bounded measurable function $h : \mathbf{E}_{\tau_n} \rightarrow \mathbb{R}$ we have

$$p = \mathbb{E} [h(\mathbf{Z}_{\tau_n})] = \iiint_{\mathbf{E}_{\tau_n} \times \dots \times \mathbf{E}_{\tau_0}} h(\mathbf{z}_{\tau_n}) \prod_{k=1}^n \mathbf{V}_k(d\mathbf{z}_{\tau_k} | \mathbf{z}_{\tau_{k-1}}) \nu_0(d\mathbf{z}_{\tau_0}). \quad (9.1)$$

In the rest of this section 2.4 we use the following notations: We denote by $\mathcal{M}(A)$ the set of bounded measurable functions on a measurable set (A, \mathcal{A}) . If f is a bounded measurable function, and η is a measure we note $\eta(f) = \int f d\eta$. If M is a Markovian kernel, we denote by $M(f)$ the function such that $M(f)(x) = \int f(y) M(dy|x)$, and for a measure η , we denote by ηM the measure such that

$$\eta M(f) = \int \int f(y) M(dy|x) \eta(dx).$$

Remember the IPS method provides an estimator of p with a different variance than the Monte-Carlo estimator. It was first introduced in [23], and with an alternative formulation in [22].

We now represent the whole method with the PDMP notation. If you feel confident with the notation change explained above we suggest the reader to go directly to section 9.2.

9.1.1 A Feynman-Kac model

The IPS method relies on a Feynman-Kac model [21] whose main characteristics are defined in this subsection. A Feynman-Kac model is a sequence of target distributions and of propagated distributions on the sets $(\mathbf{E}_{\tau_k}, \mathcal{E}_k)$, that are defined using a series of potential function $(G_k)_{k < n}$. These distributions are linked by a selection and propagation pattern defined below.

Potential functions

For each $k < n$ we denote G_k the potential function on \mathbf{E}_{τ_k} , such that:

$$\forall \mathbf{z}_{\tau_k} \in \mathbf{E}_{\tau_k}, \quad G_k(\mathbf{z}_{\tau_k}) \geq 0. \quad (9.2)$$

Target and propagated distributions

We consider a target probability measure $\tilde{\eta}_k$ for each $k \leq n$, which is defined by:

$$\tilde{\eta}_k(d\mathbf{z}_{\tau_k}) \propto \prod_{s=0}^k G_s(\mathbf{z}_{\tau_s}) \prod_{s=0}^{k-1} \mathbf{V}_{s+1}(d\mathbf{z}_{\tau_{s+1}} | \mathbf{z}_{\tau_s}), \quad (9.3)$$

or equivalently by

$$\forall B \in \mathcal{E}_{\tau_k}, \quad \tilde{\eta}_k(B) = \frac{\mathbb{E} \left[\mathbb{1}_B(\mathbf{Z}_{\tau_k}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}{\mathbb{E} \left[\prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}. \quad (9.4)$$

Then, we define the propagated target measures η_k such that $\eta_0 = \tilde{\eta}_0$ and for $k \geq 0$, $\eta_{k+1} = \tilde{\eta}_k \mathbf{V}_{k+1}$. We have :

$$\eta_{k+1}(d\mathbf{z}_{\tau_{k+1}}) \propto \prod_{s=0}^k G_s(\mathbf{z}_{\tau_s}) \prod_{s=0}^k \mathbf{V}_{s+1}(d\mathbf{z}_{\tau_{s+1}} | \mathbf{z}_{\tau_s}), \quad (9.5)$$

or equivalently

$$\forall B \in \mathcal{E}_{\tau_{k+1}}, \quad \eta_{k+1}(B) = \frac{\mathbb{E} \left[\mathbb{1}_B(\mathbf{Z}_{\tau_{k+1}}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}{\mathbb{E} \left[\prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}. \quad (9.6)$$

For $k = 0$ we will consider that $\eta_0 = \delta_{z_0}$, but the methods would still be valid if we had $\eta_0 \neq \delta_{z_0}$.

Link between distributions

As we defined it earlier η_{k+1} is the propagation of $\tilde{\eta}_k$ as

$$\eta_{k+1} = \tilde{\eta}_k \mathbf{V}_{k+1}.$$

Let Ψ_k be the application that transforms a measure η defined on \mathbf{E}_{τ_k} into a measure $\Psi_k(\eta)$ defined on \mathbf{E}_{τ_k} and such that

$$\Psi_k(\eta)(f) = \frac{\int G_k(\mathbf{z}_{\tau_k}) f(\mathbf{z}_{\tau_k}) d\eta(\mathbf{z}_{\tau_k})}{\eta(G_k)}. \quad (9.7)$$

We say that $\Psi_k(\eta)$ gives the selection of η through the potential G_k . Notice that $\tilde{\eta}_k$ is the selection of η_k as $\tilde{\eta}_k = \Psi_k(\eta_k)$. The target distributions can therefore be built according to the following pattern:

$$\eta_k \xrightarrow{\Psi_k} \tilde{\eta}_k \xrightarrow{\cdot \mathbf{V}_{k+1}} \eta_{k+1},$$

where there are successive selection and propagation steps.

Unnormalized measures

We also define the associated unnormalized measures $\tilde{\gamma}_k$ and γ_{k+1} , such that for $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$:

$$\tilde{\gamma}_k(f) = \mathbb{E} \left[f(\mathbf{Z}_{\tau_k}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right] \quad \text{and} \quad \tilde{\eta}_k(f) = \frac{\tilde{\gamma}_k(f)}{\tilde{\gamma}_k(1)}, \quad (9.8)$$

and for $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$:

$$\gamma_{k+1}(f) = \mathbb{E} \left[f(\mathbf{Z}_{\tau_{k+1}}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right] \quad \text{and} \quad \eta_{k+1}(f) = \frac{\gamma_{k+1}(f)}{\gamma_{k+1}(1)}. \quad (9.9)$$

Denoting $f_h(\mathbf{z}_{\tau_n}) = \frac{h(\mathbf{z}_{\tau_n})}{\prod_{s=0}^{n-1} G_s(\mathbf{z}_{\tau_s})}$, notice that we have:

$$p = \gamma_n(f_h) = \eta_n(f_h) \prod_{k=0}^{n-1} \eta_k(G_k). \quad (9.10)$$

Weighted propagation

We define Q_k such that for $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$,

$$Q_k(f)(\mathbf{z}_{\tau_k}) = \int_{\mathbf{E}_{\tau_{k+1}}} f(\mathbf{z}_{\tau_{k+1}}) \mathbf{V}_k(d\mathbf{z}_{\tau_{k+1}} | \mathbf{z}_{\tau_k}) G_k(\mathbf{z}_{\tau_k}) \quad (9.11)$$

and set $Q_{k,n} = Q_k Q_{k+1} \dots Q_n$. With this notation we have

$$\forall s < k, \quad \gamma_k(f) = \gamma_s Q_{s,k}(f) = \gamma_0 Q_{0,k}(f) = \eta_0 Q_{0,k}(f). \quad (9.12)$$

9.1.2 The IPS algorithm and its estimators

The IPS method provides an algorithm to generate weighted samples which approximate the probability measures η_k and $\tilde{\eta}_k$ respectively for each step k . For the sample approximating η_k , we denote $\mathbf{Z}_{\tau_k}^j$ the j^{th} trajectory and W_k^j its weight. Respectively, for the sample approximating $\tilde{\eta}_k$, we denote $\tilde{\mathbf{Z}}_{\tau_k}^j$ the j^{th} trajectory and \tilde{W}_k^j its associated weight. For simplicity reasons, in this algorithm, we consider that the samples all contain N trajectories, but it is possible to modify the sample size at each step, as illustrated in [38]. The empirical approximations of η_k and $\tilde{\eta}_k$ are denoted by η_k^N and $\tilde{\eta}_k^N$ and are defined by:

$$\tilde{\eta}_k^N = \sum_{i=1}^N \tilde{W}_k^i \delta_{\tilde{\mathbf{z}}_{\tau_k}^i} \quad \text{and} \quad \eta_k^N = \sum_{i=1}^N W_k^i \delta_{\mathbf{z}_{\tau_k}^i}. \quad (9.13)$$

So for all $k \leq n$ and $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$,

$$\tilde{\eta}_k^N(f) = \sum_{i=1}^N \tilde{W}_k^i f(\tilde{\mathbf{z}}_{\tau_k}^i) \quad \text{and} \quad \eta_k^N(f) = \sum_{i=1}^N W_k^i f(\mathbf{z}_{\tau_k}^i). \quad (9.14)$$

By plugging these estimations into equations (9.8) and (9.8), we get estimations for the unnormalized distributions. Denoting by $\tilde{\gamma}_k^N$ and γ_k^N these estimations, for all $k \leq n$ and $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$, we have:

$$\tilde{\gamma}_k^N(f) = \tilde{\eta}_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s) \quad \text{and} \quad \gamma_k^N(f) = \eta_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s). \quad (9.15)$$

Plugging the estimations η_k^N into equation (9.10), we get an estimator \hat{p} of p defined by:

$$\hat{p} = \eta_n^N(f_h) \prod_{k=0}^{n-1} \eta_k^N(G_k). \quad (9.16)$$

The algorithm building the samples is presented in figure 9.1. It follows the same logic as the algorithm that we presented in 2.4.2, except notations are adapted to the PDMP case.

We keep the following assumption: $\exists \varepsilon_1, \varepsilon_2 \in \mathbb{R}^+$ such that $\forall \mathbf{z}_{\tau_k} \in \mathbf{E}_{\tau_k}$:

$$\varepsilon_1 > G_k(\mathbf{z}_{\tau_k}) > \varepsilon_2 > 0, \quad (\text{G-PDMP})$$

so the following theorems still apply:

Theorem 22. *When (G) is verified the estimator (9.16) is unbiased and strongly consistent.*

The proof of theorem 22 can be found in [21] chapter 7.

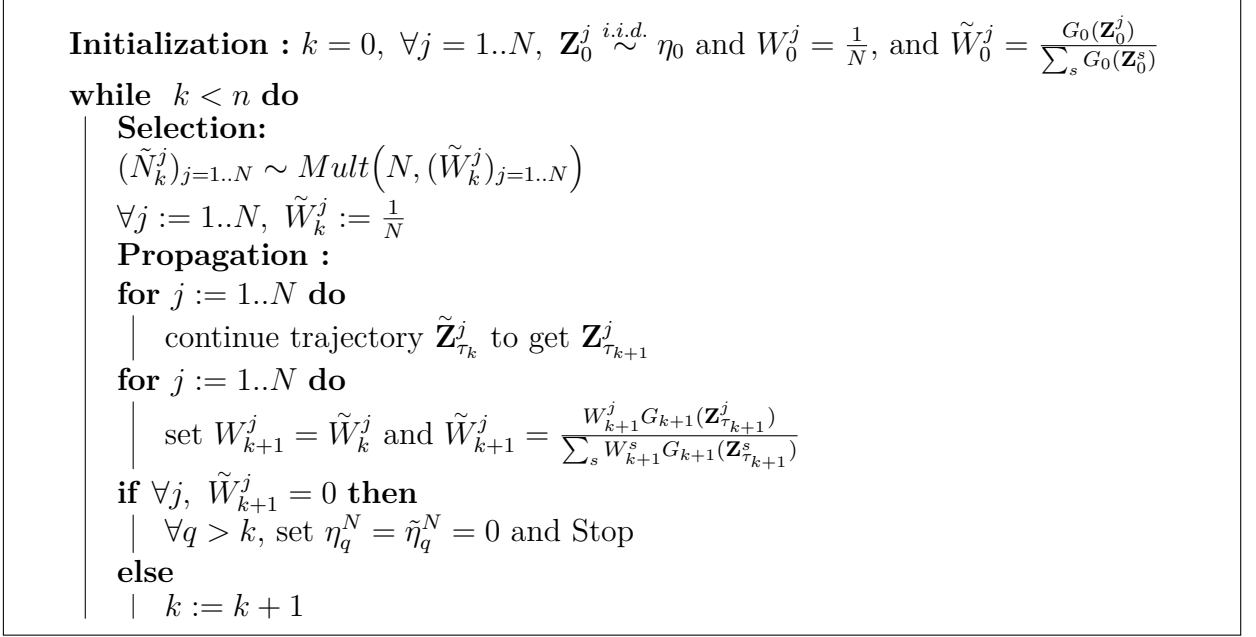


Figure 9.1 – IPS algorithm

Theorem 23. *When (G) is verified:*

$$\sqrt{N}(\hat{p} - p) \xrightarrow[N \rightarrow \infty]{d} \mathcal{N}(0, \sigma_{IPS,G}^2), \quad (9.17)$$

where

$$\sigma_{IPS,G}^2 = \sum_{k=0}^{n-1} \gamma_k(1)^2 \eta_k \left(\left[Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h) \right]^2 \right) \quad (9.18)$$

$$= \sum_{k=0}^{n-1} \left\{ \mathbb{E}_{z_0} \left[\prod_{i=0}^{k-1} G_i(\mathbf{Z}_{\tau_i}) \right] \mathbb{E}_{z_0} \left[\mathbb{E}[h(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_k}]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_{\tau_s}) \right] - p^2 \right\}. \quad (9.19)$$

A proof of this CLT can be found in [21] chapter 9.

9.1.3 Variance estimation for the PDMP case

For the estimation of the variance $\sigma_{IPS,G}^2$, the two estimators presented in [38] and subsection 2.4.3 can also be transposed to the PDMP case. Let A_k^j be the ancestor index of the j^{th} trajectory in the k -th selected sample, such that $\tilde{\mathbf{Z}}_{\tau_k}^j = \mathbf{Z}_{\tau_k}^{A_k^j}$. Let e_n^i be the Eve index of i , that is defined recursively by:

$$e_0^i = i \text{ and for } k, 0 < k \leq n, \quad e_k^i = A_k^{e_{k-1}^i}. \quad (9.20)$$

So e_k^i is the ancestor index at step 0 of the i -th particle of the k -th sample.

The first estimator of the variance is then defined by:

$$\hat{\sigma}_1^2 = \gamma_n^N (1)^2 \left\{ \eta_n^N (f_h)^2 - \frac{N^{n-1}}{(N-1)^{n+1}} \sum_{i,j: e_n^i \neq e_n^j} f_h(\mathbf{Z}_{\tau_n}^i) f_h(\mathbf{Z}_{\tau_n}^j) \right\}. \quad (9.21)$$

It satisfies the following theorem:

Theorem 24. $\hat{\sigma}_1^2$ is an unbiased and weakly convergent estimator of $\sigma_{IPS,G}^2$.

The second estimator relies on a rewriting of (9.19) in (9.23). For a given vector $b \in \{0, 1\}^n$ and function $f \in \mathbb{R}^{\mathbf{E}_{\tau_n}^2}$ we introduce the quantity $\mu_b(f)$ defined by:

$$\mu_b(f) = \mathbb{E}^b \left[\prod_{k=0}^{n-1} G_k(\mathbf{Y}_{\tau_k}) G_k(\mathbf{Y}'_{\tau_k}) f(\mathbf{Y}_{\tau_n}, \mathbf{Y}'_{\tau_n}) \right], \quad (9.22)$$

Where the random trajectories $(\mathbf{Y}_{\tau_k})_{k \leq n}$ with values in (E, \mathcal{E}) are such that $\mathbf{Y}_{\tau_0} \sim \eta_0$ and for $k > 0$ $\mathbf{Y}_{\tau_k} \sim \mathbf{V}_k(\cdot | \mathbf{Y}_{\tau_{k-1}})$. The second random vector $(\mathbf{Y}'_{\tau_k})_{k \leq n}$ is such that, if $b_0 = 0$ $\mathbf{Y}'_{\tau_0} \sim \eta_0$ with \mathbf{Y}'_{τ_0} independent of \mathbf{Y}_{τ_0} , and if $b_0 = 1$ set $\mathbf{Y}'_{\tau_0} = \mathbf{Y}_{\tau_0}$. For $k \leq 0$ if $b_{k+1} = 0$ take $\mathbf{Y}'_{\tau_{k+1}} \sim \mathbf{V}_{k+1}(\cdot | \mathbf{Y}'_{\tau_k})$ with $\mathbf{Y}'_{\tau_{k+1}}$ independent of $\mathbf{Y}_{\tau_{k+1}}$, and if $b_k = 1$ set $\mathbf{Y}'_{\tau_{k+1}} = \mathbf{Y}_{\tau_{k+1}}$. Note the expectation \mathbb{E}^b now includes the vector b used to build $(\mathbf{Y}'_{\tau_k})_{k \leq n}$. For a function $f \in \mathbb{R}^{\mathbf{E}_{\tau_n}^2}$ we define $f^{\otimes 2}$ as the function of $\mathbb{R}^{\mathbf{E}_{\tau_n}^2}$ such that $f^{\otimes 2}(\mathbf{Y}_{\tau_n}, \mathbf{Y}'_{\tau_n}) = f(\mathbf{Y}_{\tau_n}) f(\mathbf{Y}'_{\tau_n})$. Denoting by e_k the vector with the $k+1^{\text{th}}$ coordinate equal to 1 and null coordinates everywhere else, and denoting by 0_n the null vector of size n , we can rewrite the variance as follows:

$$\sigma_{IPS,G}^2 = \sum_{k=0}^{n-1} \mu_{e_k}(f_h^{\otimes 2}) - \mu_{0_n}(f_h^{\otimes 2}) \quad (9.23)$$

In order to estimate the variance the authors of [38] provide an estimator $\mu_b^N(f^{\otimes 2})$ of the quantity $\mu_b(f^{\otimes 2})$ for any $b \in \{0, 1\}^n$ and set

$$\sigma_2^2 = \sum_{k=0}^{n-1} \mu_{e_k}^N(f_h^{\otimes 2}) - \mu_{0_n}^N(f_h^{\otimes 2}). \quad (9.24)$$

In order to estimate the quantities $\mu_b(f^{\otimes 2})$ they introduce two auxiliary random vectors. These vectors are drawn from the samples built with the IPS algorithm. They are denoted $\mathbf{Z}^{L^1} = (\mathbf{Z}_{\tau_0}^{L_0^1}, \dots, \mathbf{Z}_{\tau_n}^{L_n^1})$ and $\mathbf{Z}^{L^2} = (\mathbf{Z}_{\tau_0}^{L_0^2}, \dots, \mathbf{Z}_{\tau_n}^{L_n^2})$, where $L^1 = (L_0^1, \dots, L_n^1)$ and $L^2 = (L_0^2, \dots, L_n^2)$ represent the indices of the trajectories picked in the samples. These vectors of indices are associated with the genealogical structure of the samples with the algorithm. L^1 traces backward the ancestral lineage of a trajectory. L^2 traces backward the ancestral lineage of a trajectory that breaks each time it touches the L^1 lineage. They are built as follows: L_n^1 is drawn uniformly from $\{1, \dots, N\}$, and $\forall k < n$, we set $L_k^1 = A_k^{L_{k+1}^1}$. L_n^2 is also drawn uniformly from $\{1, \dots, N\}$, but $\forall p < n$, if $L_{k+1}^2 \neq L_{k+1}^1$, we

set $L_k^2 = A_k^{L_{k+1}^2}$, and when $L_{k+1}^2 = L_{k+1}^1$ we draw L_k^2 from $\{1, \dots, N\}$ with a probability $\mathbb{P}(L_k^2 = j) = \frac{W_k^j G(\mathbf{Z}_{\tau_k}^j)}{\sum_{s=1}^N W_k^s G(\mathbf{Z}_{\tau_k}^s)}$.

Denoting $\underline{A} = \{A_k^j \mid 0 \leq k \leq n, 1 \leq j \leq N\}$ the set of all ancestors, and $\underline{\mathbf{Z}} = \{\mathbf{Z}_{\tau_k}^j \mid 0 \leq k \leq n, 1 \leq j \leq N\}$ the set of all the sampled trajectories, and I_b the set of pairs of lineage such that $L_k^1 = L_k^2$ if and only if $b_k = 1$, $\mu_b(f)$ is estimated by:

$$\mu_b^N(f) = \left(\prod_{k=0}^n \frac{N}{(N-1)^{1-b_k}} \right) \gamma_n^N(1)^2 \mathbb{E} \left[\mathbb{1}_{I_b}(L^1, L^2) f(\mathbf{Z}_{\tau_n}^{L^1}, \mathbf{Z}_{\tau_n}^{L^2}) \mid \underline{A}, \underline{\mathbf{Z}} \right]. \quad (9.25)$$

Theorem 25. $\mu_b^N(f)$ is an unbiased and weakly convergent estimator of $\mu_b(f)$.

Therefore we have the following theorem:

Theorem 26. $\hat{\sigma}_2^2$ is an unbiased and weakly convergent estimator of $\sigma_{IPS,G}^2$.

The proof for both these theorems can be found in the supplement of [38].

9.1.4 The SMC improvement for PDMP

Similarly to what was presented in subsection 2.4.4, in order to avoid pointless resampling, one can trigger the selection step only when the weights are unbalanced. This is done in the Sequential Monte Carlo (SMC) algorithm with adaptive resampling presented in Figure 9.2. In this algorithm, the heterogeneity of the weights is quantified using the effective sample size. At the k^{th} step the effective sample size is defined by:

$$ESS_k = \frac{\left(\sum_{j=1}^N W_k^j G_k(\mathbf{Z}_{\tau_k}^j) \right)^2}{\sum_{i=1}^N \left(W_k^i G_k(\mathbf{Z}_{\tau_k}^i) \right)^2}. \quad (9.26)$$

Its value is between 1 and N and it measures the homogeneity in the candidate weights $\frac{W_k^i G_k(\mathbf{Z}_{\tau_k}^i)}{\sum_j W_k^j G_k(\mathbf{Z}_{\tau_k}^j)}$: when $ESS_k = N$ the weights are perfectly balanced and are all equal to $\frac{1}{N}$, and conversely when $ESS_k = 1$ all the weights are null except one, which concentrates the totality of the mass. Therefore, one considers the weights are too unbalanced when $ESS_k < eN$ where $e \in [0, 1]$ is a tuning parameter.

Other improvements of the IPS, like those mentioned in subsection 2.4.4, are compatible with PDMP.

```

Initialization :  $k = 0$ ,  $\forall i := 1..N, \mathbf{Z}_0^i = (z_0)$  and  $W_0^i = \frac{1}{N}$ , and  $\tilde{W}_0^i = \frac{G_0(\mathbf{Z}_0^i)}{\sum_j G_0(\mathbf{Z}_0^j)}$ 
while  $k < n$  do
    Selection:
    if  $ESS_k < eN$  then
         $(\tilde{N}_k^j)_{j=1..N} \sim Mult(N, (\tilde{W}_k^j)_{j=1..N})$  and set  $\forall i = 1..N, \tilde{W}_k^i := \frac{1}{N}$ 
    else
        for  $i := 1..N$  do
            set  $\tilde{\mathbf{Z}}_{\tau_k}^i := \mathbf{Z}_{\tau_k}^i$ 
    Propagation :
    for  $i := 1..N$  do
        continue trajectory  $\tilde{\mathbf{Z}}_{\tau_k}^i$  to get  $\mathbf{Z}_{\tau_{k+1}}^i$ 
    for  $i := 1..N$  do
        set  $W_{k+1}^i = \tilde{W}_k^i$  and  $\tilde{W}_{k+1}^i = \frac{W_{k+1}^i G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^i)}{\sum_j W_{k+1}^j G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^j)}$ 
    if  $\forall j, \tilde{W}_{k+1}^j = 0$  then
         $\forall q > k$ , set  $\eta_q^N = \tilde{\eta}_q^N = 0$  and Stop
    else
         $k := k + 1$ 
    
```

Figure 9.2 – SMC algorithm with adaptive resampling steps for PDMP

9.2 Concentrated PDMP make the IPS inefficient

9.2.1 The kind of PDMP used in the reliability analysis

Low jump rate

For reliability assessment of a highly reliable system one often models the system by a PDMP with low jump rates. Indeed, the possible spontaneous transitions are often failures or repairs, and the components of the system are reliable: their failures are rare and their repairs take time, hence the low jump rates.

Discrete jump kernel

Although in a PDMP the jumps model transitions from state to state (z^- to z^+), we here only consider transitions from mode to mode (m^- to m^+). The reason is that the evolution of the physical variables during a jump is often deterministic once we know the arrival statuses of the components. We consider that as soon as the mode m^+ is known, the only option for the physical variables is to arrive in a point denoted $x_{z \rightarrow m^+}$. Therefore ν_{z^-} the reference measure for the jump kernel is taken to be equal to:

$$\forall B \in \mathcal{B}(E), \quad \nu_{z^-}(B) = \sum_{m^+ \in \mathbb{M}} \delta_{(x_{z \rightarrow m^+}, m^+)}(B), \quad (9.27)$$

and the jump kernel is such that :

$$\forall B \in \mathcal{B}(E), \quad \mathcal{K}_{z^-}(B) = \sum_{m^+ \in \mathbb{M}} \mathbb{P}(Z_T = (x_{z \rightarrow m^+}, m^+) | Z_T^- = z^-) \delta_{(x_{z \rightarrow m^+}, m^+)}(B), \quad (9.28)$$

The jump Kernel is therefore a discrete law of probability.

Concentrated jump kernel on the boundaries

The boundaries are used to model control mechanisms, and the jump kernel on boundaries gives the probability for each possible output of the control mechanism triggered. As failures on demand during a control mechanism are unlikely the jump kernels on boundaries are concentrated on one safe arrival state (i.e. the state aimed by the control mechanism).

9.2.2 Concentrated PDMP

Definition 36. For an interval I , the PDMP $(Z_s)_{s \in I}$ is said to be a *concentrated PDMP* if there exists a trajectory $\mathbf{a} \in \mathbf{E}_I$ such that

$$\mathbb{P}(\mathbf{Z}_I = \mathbf{a}) \simeq 1 \quad (9.29)$$

In reliability analysis, the systems are often modeled by concentrated PDMPs.

The preponderant trajectory

In reliability analysis the trajectory \mathbf{a} is often the trajectory without failure nor repair. As jump rates are low the probability of not having a spontaneous jump is close to one. For instance at a $k + 1$ -th jump this probability verifies

$$\mathbb{P}_{z_{s_k}}(T_k = t_{z_{s_k}}^*) = \exp \left[-\Lambda_{z_{s_k}}(t_{z_{s_k}}^*) \right] \simeq 1.$$

So only jumps on boundaries are likely. When the process hits a boundary ∂E_m , remember the reference measure of the jump is discrete, see equation (9.28). At this point there is a high probability that the arrival state is the state aimed by the control mechanism. Denoting by z_{s_k} this state for a k -th jump, we have :

$$K_{z_{s_k}^-}(z_{s_k}) = \mathcal{K}_{z_{s_k}^-}(\{z_{s_k}\}) \simeq 1.$$

So if \mathbf{z}_t is a trajectory with no failure and no repair of reasonable size we have:

$$\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) = \prod_{k=0}^n \exp \left[-\Lambda_{z_{s_k}}(t_{z_{s_k}}^*) \right] \prod_{k=1}^n \mathcal{K}_{z_{s_k}^-}(\{z_{s_k}\}) \simeq 1. \quad (9.30)$$

In the IPS method, in order to simulate trajectories on the interval $[\tau_0, \tau_n]$ we simulate the trajectories sequentially by simulating on the successive intervals $(\tau_k, \tau_{k+1}]$ where k range from 0 to $n - 1$. As these intervals of time are relatively small the PDMP modeling the state of the system is often a concentrated PDMP on $(\tau_k, \tau_{k+1}]$.

9.2.3 A poor empirical approximation of the propagated distributions within the IPS

When it is used on a reliable system and therefore on a concentrated PDMP (see the previous Section 9.2.2), the IPS method tends to loose in efficiency. This efficiency loss can be attributed to the exploration steps. Remember that an exploration step comes after a selection step: it builds a sample $(\mathbf{Z}_{\tau_{k+1}}^j, W_{k+1}^j)_{j \leq N}$ by extending the trajectories of a selected sample $(\tilde{\mathbf{Z}}_{\tau_k}^j, \tilde{W}_k^j)_{j \leq \tilde{N}}$. This newly built sample $(\mathbf{Z}_{\tau_{k+1}}^j, W_{k+1}^j)_{j \leq N}$ fulfills two goals : 1) It contributes to the empirical approximation η_{k+1}^N of η_{k+1} . 2) It is used as a candidate sample for the next selection. But this second goal is often poorly achieved with a concentrated PDMP.

Indeed, in order to get a good approximation $\tilde{\eta}_{k+1}^N$ of $\tilde{\eta}_{k+1}$, it is preferable that the candidate sample to selection $(\mathbf{Z}_{\tau_{k+1}}^j, W_{k+1}^j)_{j \leq N}$ contains as many different trajectories as possible, along with high potential trajectories. Unfortunately, with this kind of PDMP, it is generally not the case: the candidate sample often contains several replicates of the same trajectories, and no high potential trajectory. Therefore each distribution $\tilde{\eta}_k$ is poorly represented, and so is each target distribution η_{k+1} , which eventually deteriorates the quality of the estimator \hat{p} .

To understand why the exploration steps are not likely to generate many different trajectories with a concentrated PDMP, we have to come back at the beginning of the propagation step. At that point, the sample $(\tilde{\mathbf{Z}}_{\tau_k}^j, \tilde{W}_k^j)_{1 \leq j \leq N}$ is naturally clustered because of the previous selection step, each of the clusters containing several replicates of the same trajectories. We can rewrite (9.13) in the following way

$$\tilde{\eta}_k^N = \sum_{i=1}^N \tilde{W}_k^i \delta_{\tilde{\mathbf{Z}}_{\tau_k}^i} = \frac{1}{N} \sum_{j=1}^N \tilde{N}_k^j \delta_{\mathbf{Z}_{\tau_k}^j} \quad (9.31)$$

where $\sum_{j=1}^N \tilde{N}_k^j = N$. In practice many of the \tilde{N}_k^j are null and only a few are positive and the N resampled trajectories $(\tilde{\mathbf{Z}}_{\tau_k}^j)_{j \leq N}$ are concentrated on a few trajectories. Then, each of the \tilde{N}_k^j trajectories of the j -th cluster is extended by using the same distribution $\mathbf{V}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$. (For each index i such that $A_k^i = j$ the trajectory $\tilde{\mathbf{Z}}_{\tau_k}^i$ is extended with the kernel $\mathbf{V}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$). As the kernel $\mathbf{V}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$ corresponds to a concentrated PDMP, it is likely to extend all the trajectories of a cluster in the same manner. The trajectory $\mathbf{a}_{\tau_{k+1}}^{k,j}$ which extends $\mathbf{Z}_{\tau_k}^j$ until τ_{k+1} without spontaneous jump or failure concentrates the mass

of the kernel $\mathbf{V}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$. Indeed, at this point we have :

$$\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) = \mathbb{P}(\mathbf{Z}_{\tau_{k+1}} = \mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k} = \mathbf{Z}_{\tau_k}^j) \simeq 1. \quad (9.32)$$

Therefore each of the trajectories $\tilde{\mathbf{Z}}_{\tau_k}^i = \mathbf{Z}_{\tau_k}^j$ in a cluster tends to be extended into $\mathbf{a}_{\tau_{k+1}}^{k,j}$. Thus, the trajectories within a cluster are likely to stay clumped together during the propagation, and the propagated sample $(\mathbf{Z}_{\tau_{k+1}}^j, W_{k+1}^j)_{j \leq N}$ is very likely to be clustered too. When the preponderant trajectories $\mathbf{a}_{\tau_{k+1}}^{k,j}$ have low potential values, the sample is not likely to contain high potential trajectories. Consequently the selection step having no good candidates and too few candidates, it tends to yield an inaccurate estimation of the distributions $\tilde{\eta}_k$.

This situation is typical of reliability assessment. In that context, a well constructed potential function is close to G_k^* wherein $h = \mathbb{1}_{\mathcal{O}}$. So the potential of a trajectory $G_{k+1}(\mathbf{Z}_{\tau_{k+1}})$ should be high if its final state $Z_{\tau_{k+1}}$ is more degraded than the state Z_{τ_k} . This generally implies that $\mathbf{Z}_{\tau_{k+1}}$ includes at least one component failure between τ_k and τ_{k+1} . As the preponderant trajectories $\mathbf{a}_{\tau_{k+1}}^{k,j}$ do not contain failure between τ_k and τ_{k+1} they generally are associated with low potential values.

This phenomenon is due to the fact that the distribution of $\mathbf{z}_{\tau_{k+1}}$ can be decomposed into a Dirac measure in $\mathbf{a}_{\tau_{k+1}}^{k,j}$ with a high probability $\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)$ and a continuous part. As the segment of the trajectory $\mathbf{a}_{\tau_{k+1}}^{k,j}$ on $(\tau_k, \tau_{k+1}]$ relates to a Dirac contribution of the measure $\zeta_{Z_{\tau_k}^j, \tau_{k+1} - \tau_k}$, we can decompose the expected propagation of the trajectory $\tilde{\mathbf{Z}}_{\tau_k}^j$ in this way:

$$\begin{aligned} \delta_{\mathbf{Z}_{\tau_k}^j} \mathbf{V}_{k+1}(f) &= f(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}) \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) + \int_{\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}} f(\mathbf{z}_{\tau_{k+1}}) \mathbf{V}_{k+1}(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j), \\ &= f(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}) v_{k+1}(\mathbf{a}_{\tau_{k+1}, (\tau_k, \tau_{k+1})}^{k,j} | \mathbf{Z}_{\tau_k, (\tau_{k-1}, \tau_k)}^j) \\ &\quad + \int_{\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}} f(\mathbf{z}_{\tau_{k+1}}) v_{k+1}(d\mathbf{z}_{\tau_{k+1}, (\tau_k, \tau_{k+1})} | \mathbf{Z}_{\tau_k, (\tau_{k-1}, \tau_k)}^j) \delta_{Z_{\tau_k}^j}(d\mathbf{z}_{\tau_k}) \quad (9.33) \end{aligned}$$

where $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$. And the expected propagation of η_k^N would be:

$$\begin{aligned} \tilde{\eta}_k^N \mathbf{V}_{k+1}(f) &= \sum_{j=1}^N \frac{\tilde{N}_k^j}{N} \delta_{\mathbf{Z}_{\tau_k}^j} \mathbf{V}_{k+1}(f) \\ &= \sum_{j=1}^N \frac{\tilde{N}_k^j}{N} f(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}) \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) \\ &\quad + \frac{\tilde{N}_k^j}{N} \int_{\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}} f(\mathbf{z}_{\tau_{k+1}}) \mathbf{V}_{k+1}(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j) \\ &\simeq \sum_{j=1}^N \frac{\tilde{N}_k^j}{N} f(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}), \quad (9.34) \end{aligned}$$

because the probability $\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}|\mathbf{Z}_{\tau_k}^j) \simeq 1$. The consequence is the set $\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}$ is not likely to be explored empirically, and unfortunately this set is where the high potential trajectories are in our case. The issue of the IPS with concentrated PDMPs is that, in practice, this set is not represented empirically in the samples of the algorithm, which deteriorates the estimates of the target densities.

Our solution is to modify the propagation step, using the memorization method presented in the next chapter because this method enforces the simulation of trajectories in this set.

Efficient generation of the trajectories using the Memorization method

This chapter presents the memorization method that was first introduced in [36]. Remember that we consider that a trajectory \mathbf{a}_t is preponderant whenever $p_{\mathbf{a}_t} = \mathbb{P}(\mathbf{Z}_t = \mathbf{a}_t) > 0$. Assuming we know such a preponderant trajectory \mathbf{a}_t , the memorization method allows to generate a trajectory \mathbf{Z}_t which differs from this preponderant trajectory \mathbf{a}_t .

10.1 Advantage of Memorization over a rejection algorithm

The interest of the method, compared to a rejection algorithm, is that we generate a trajectory $\mathbf{Z}_t \neq \mathbf{a}_t$ in one shot, whereas a rejection algorithm may generate several times the preponderant trajectory \mathbf{a}_t before generating a trajectory different from \mathbf{a}_t . This is especially interesting when the probability $p_{\mathbf{a}_t} = \mathbb{P}(\mathbf{Z}_t = \mathbf{a}_t)$ is close to 1, as, with a rejection algorithm, the average number of tries to get a trajectory different from \mathbf{a}_t would be $\frac{1}{1-p_{\mathbf{a}_t}}$ which is then very high. Therefore with a rejection algorithm much computational effort would be wasted generating \mathbf{a}_t over and over.

10.2 Principle of the memorization method

Work with the differentiation time

The key idea of the memorization method is to consider the stopping time τ defined such that:

$$\forall s < \tau, \quad Z_s = a_s \quad \text{and} \quad Z_\tau \neq a_\tau. \quad (10.1)$$

This time τ is the time at which the trajectory \mathbf{Z}_t differentiates itself from \mathbf{a}_t . So, to generate \mathbf{Z}_t knowing $\tau \leq t$ is equivalent to generate \mathbf{Z}_t knowing it differs from \mathbf{a}_t . In order to simulate a trajectory \mathbf{Z}_t avoiding \mathbf{a}_t , one can follow these three steps:

1. generate τ knowing $\tau \leq t$, and set $\mathbf{Z}_{\tau-} = \mathbf{a}_{\tau-}$,

2. generate Z_τ knowing $Z_\tau \neq a_\tau$,
3. generate the rest of the trajectory normally until t .

These steps are not difficult to realize, except for the first one.

Generate τ knowing $\tau \leq t$

To achieve this first step, the authors in [36] propose to generate τ knowing $\tau \leq t$ by using a method equivalent to the inverse transform sampling method. We present hereafter the theoretical foundation for this method. We denote by F the cumulative distribution function (cdf) of τ knowing $\tau \leq t$:

$$F(v) = \mathbb{P}(\tau < v | \tau \leq t), \quad (10.2)$$

and we denote by F^{-1} its generalized inverse defined by

$$F^{-1}(x) = \inf_{v>0} \{v \mid F(v) \geq x\}. \quad (10.3)$$

We also denote by \tilde{F} the function defined by

$$\tilde{F}(v) = \mathbb{P}(\mathbf{Z}_{v^-} = \mathbf{a}_{v^-}) = \prod_{k=0}^{n(\mathbf{a}_v)} \exp \left[-\Lambda_{a_{s_k}}(t_k) \right] \prod_{k=1}^{n(\mathbf{a}_v)} \left(\mathcal{K}_{a_{s_k}^-}(\{a_{s_k}\}) \right)^{\mathbb{1}_{t_k > 0}} \quad (10.4)$$

where $\Theta_v(\mathbf{a}_v) = ((a_{s_k}, t_k))_{0 \leq k \leq n(\mathbf{a}_v)}$. Note that \tilde{F} is discontinuous in each jump time s_k where $\mathcal{K}_{a_{s_k}^-}(\{a_{s_k}\}) \neq 1$, so the inverse of \tilde{F} is not necessarily defined everywhere on $[p_{\mathbf{a}_t}, 1]$. For this reason we consider \tilde{F}^{-1} , the generalized inverse of \tilde{F} defined by

$$\tilde{F}^{-1}(x) = \sup_{v>0} \{v \mid \tilde{F}(v) \leq x\}. \quad (10.5)$$

\tilde{F}^{-1} extends the inverse of \tilde{F} constantly where it is not defined, this extension being done from the left so that \tilde{F}^{-1} is right continuous.

The inverse transform sampling method consists in generating $U \sim Unif(0, 1)$ and taking $F^{-1}(U)$ as a realization of $\tau \mid \tau \leq t$. The simulation of such random variables is also presented in [24]. Now, note that the expression of the cdf F can be related to \tilde{F} , indeed we have:

$$\forall v < t, \quad F(v) = \frac{\mathbb{P}(\tau < v)}{\mathbb{P}(\tau \leq t)} = \frac{1 - \mathbb{P}(\tau \geq v)}{1 - \mathbb{P}(\tau > t)} = \frac{1 - \mathbb{P}(\mathbf{Z}_{v^-} = \mathbf{a}_{v^-})}{1 - \mathbb{P}(\mathbf{Z}_t = \mathbf{a}_t)} = \frac{1 - \tilde{F}(v)}{1 - p_{\mathbf{a}_t}}.$$

Consequently we have that

$$\begin{aligned}
 F^{-1}(U) &= \inf_{v>0} \{v \mid F(v) \geq U\} \\
 &= \sup_{v>0} \left\{v \mid \tilde{F}(v) \leq 1 - U(1 - p_{\mathbf{a}_t})\right\}, \\
 &= \tilde{F}\left(1 - U(1 - p_{\mathbf{a}_t})\right).
 \end{aligned} \tag{10.6}$$

Also, as U has uniform distribution on $[0, 1]$, $\tilde{U} = 1 - U(1 - p_{\mathbf{a}_t})$ is uniform on $[p_{\mathbf{a}_t}, 1]$. Therefore, sampling with the inverse transform method is equivalent to simulating $\tilde{U} \sim \text{Unif}(p_{\mathbf{a}_t}, 1)$ and taking $\tilde{F}^{-1}(\tilde{U})$ as a realization of $\tau \mid \tau \leq t$.

Assuming we first generate the trajectory \mathbf{a}_t and generate \tilde{U} according to a uniform distribution on $(p_{\mathbf{a}_t}, 1)$, we now show how to evaluate $\tilde{F}^{-1}(\tilde{U})$. We consider that during the generation of \mathbf{a}_t , we computed and memorized $\mathbb{P}(\mathbf{Z}_{s_k^-} = \mathbf{a}_{s_k^-})$ and $\mathbb{P}(\mathbf{Z}_{s_k} = \mathbf{a}_{s_k})$ for each jump in the trajectory, and also for $\mathbb{P}(\mathbf{Z}_t = \mathbf{a}_t)$. Then we distinguish two cases, either there exists $k \leq n(\mathbf{a}_t)$ such that $\mathbb{P}(\mathbf{Z}_{s_k^-} = \mathbf{a}_{s_k^-}) \geq \tilde{U} > \mathbb{P}(\mathbf{Z}_{s_k} = \mathbf{a}_{s_k})$, either there exists $k \leq n(\mathbf{a}_t)$ such that $\mathbb{P}(\mathbf{Z}_{s_k} = \mathbf{a}_{s_k}) \geq \tilde{U} > \mathbb{P}(\mathbf{Z}_{s_{k+1}^-} = \mathbf{a}_{s_{k+1}^-})$ where we take the convention that $s_{n(\mathbf{a}_t)+1}^- = t$. The first case is quite simple as by definition of \tilde{F}^{-1} we get $\tilde{F}^{-1}(\tilde{U}) = s_k$. In the second case, \tilde{F} being continuous and strictly decreasing on $[s_k, s_{k+1})$, it is invertible on this interval, and \tilde{F}^{-1} corresponds to \tilde{F} 's inverse on $(\tilde{F}(s_{k+1}^-), \tilde{F}(s_k)]$. So $\tilde{F}^{-1}(\tilde{U}) \in [s_k, s_{k+1})$ and $\tilde{F}(\tilde{F}^{-1}(\tilde{U})) = \tilde{U}$. Notice that

$$\forall v \in [s_k, s_{k+1}), \quad \tilde{F}(v) = \tilde{F}(s_k) \times \exp\left[-\Lambda_{a_{s_k}}(v - s_k)\right]. \tag{10.7}$$

So in particular, for $v = \tilde{F}^{-1}(\tilde{U})$, we have :

$$\tilde{U} = \tilde{F}(\tilde{F}^{-1}(\tilde{U})) = \tilde{F}(s_k) \times \exp\left[-\Lambda_{a_{s_k}}(\tilde{F}^{-1}(\tilde{U}) - s_k)\right], \tag{10.8}$$

or equivalently

$$\log\left(\frac{\tilde{F}(s_k)}{\tilde{U}}\right) = \int_0^{\tilde{F}^{-1}(\tilde{U}) - s_k} \lambda_{a_{s_k}}(u) du. \tag{10.9}$$

To determine $\tilde{F}^{-1}(\tilde{U})$ we look for the value s such that the integral $\int_0^s \lambda_{a_{s_k}}(u) du$ is equal to $\log\left(\frac{\tilde{F}(s_k)}{\tilde{U}}\right)$ by dichotomy, then we set $\tilde{F}^{-1}(\tilde{U}) = s_k + s$.

To sum up the generation of a realization of $\tau \mid \tau \leq t$ we proceed as follows:

1. Generate $\tilde{U} \sim \text{Unif}(p_{\mathbf{a}_t}, 1)$, and set $k = 0$

2. If $\mathbb{P}(\mathbf{Z}_{s_k} = \mathbf{a}_{s_k}) \geq \tilde{U} > \mathbb{P}(\mathbf{Z}_{s_{k+1}} = \mathbf{a}_{s_{k+1}})$, then find $s \in [0, s_{k+1} - s_k)$ such that

$$\log \left(\frac{\tilde{F}(s_k)}{\tilde{U}} \right) = \int_0^s \lambda_{a_{s_k}}(u) du,$$

and set $\tau = s_k + s$.

3. If $\mathbb{P}(\mathbf{Z}_{s_{k+1}} = \mathbf{a}_{s_{k+1}}) \geq \tilde{U} > \mathbb{P}(\mathbf{Z}_{s_{k+1}} = \mathbf{a}_{s_{k+1}})$, then $\tau = s_{k+1}$
4. If the condition above is not satisfied, set $k = k + 1$, if $k \leq n(\mathbf{a}_s)$ repeat the steps 2 to 4

The IPS+M method for concentrated PDMPs

11.1 Modify the propagation of clusters

We have seen in section 9.2.3 that, when they are applied on a concentrated PDMP, the exploration steps of the IPS method tend to simulate the same trajectory, which reduces the efficiency of the exploration and consequently it reduces the efficiency of the IPS method. In order to diversify the simulated trajectories, and to increase the precision of the estimation, we propose to modify the exploration steps, by modifying the way we extend the selected trajectories. Here we consider that the size of the propagated sample can differ from the size of the previous selected sample. We now denote \tilde{N}_k the size of the k^{th} selected sample, and N_{k+1} the size of the k^{th} propagated sample, with the convention $N_0 = N$. We stressed out, in section 9.2.3, that the propagation step aims at providing an estimation of $\eta_{k+1} = \tilde{\eta}_k \mathbf{V}_{k+1}$ using the selected sample $(\tilde{\mathbf{Z}}_{\tau_k}^j, \tilde{W}_k^j)_{j \leq \tilde{N}_k}$. In other words, the selection step aims at providing a propagated weighted sample $(\mathbf{Z}_{\tau_{k+1}}^j, W_{k+1}^j)_{j \leq N_{k+1}}$ to estimate the distribution $\tilde{\eta}_k^{\tilde{N}_k} \mathbf{V}_{k+1}$ defined by:

$$\tilde{\eta}_k^{\tilde{N}_k} \mathbf{V}_{k+1}(f) = \sum_{j=1}^{\tilde{N}_k} \tilde{W}_k^j \delta_{\tilde{\mathbf{Z}}_{\tau_k}^j} \mathbf{V}_{k+1}(f) = \sum_{j=1}^{N_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} \delta_{\mathbf{Z}_{\tau_k}^j} \mathbf{V}_{k+1}(f), \quad (11.1)$$

where $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$. We denote by $\bar{\mathbf{V}}_{k+1}$ the Markovian kernel from \mathbf{E}_{τ_k} to $\mathbf{E}_{\tau_{k+1}}$ such that, for any trajectory $\mathbf{Z}_{\tau_k}^j$, $\bar{\mathbf{V}}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$ is the conditioning of $\mathbf{V}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$ to $\mathbf{E}_{\tau_k} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}$. $\bar{\mathbf{V}}_{k+1}$'s density with respect to $\zeta_{\tau_{k+1}}$ verifies:

$$\bar{\mathbf{V}}_{k+1}(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j) = \frac{\mathbf{V}_{k+1}(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j)}{1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)} \mathbb{1}_{\mathbf{z}_{\tau_{k+1}} \neq \mathbf{a}_{\tau_{k+1}}^{k,j}}. \quad (11.2)$$

Using (11.1) we can decompose $\tilde{\eta}_k^{\tilde{N}_k} \mathbf{V}_{k+1}$ as follows:

$$\tilde{\eta}_k^{\tilde{N}_k} \mathbf{V}_{k+1}(f) = \sum_{j=1}^{N_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} \left[\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) f(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\}) + (1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)) \delta_{\mathbf{Z}_{\tau_k}^j} \bar{\mathbf{V}}_{k+1}(f) \right]. \quad (11.3)$$

In the original IPS algorithm, the sample approximating $\tilde{\eta}_k^{\tilde{N}_k} \mathbf{V}_{k+1}$ is built by directly extending each trajectory in the selected sample. When we extend the replicates of a cluster, in average a proportion $\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)$ of the replicates are extended into $\mathbf{a}_{\tau_{k+1}}^{k,j}$. This proportion of trajectories extended into $\mathbf{a}_{\tau_{k+1}}^{k,j}$ then serves as an estimation of $\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)$. But it is not necessary to waste all these replicates to estimate the probability of the preponderant trajectory. If we use equation (11.3), we would need to generate the trajectory $\mathbf{a}_{\tau_{k+1}}^{k,j}$ only once to assess its contribution to the propagation of the cluster. Also, $\mathbf{a}_{\tau_{k+1}}^{k,j}$ is easy to get. To generate it, it generally suffices to run the simulation process starting from the state $\mathbf{Z}_{\tau_k}^j$ until time τ_{k+1} , while setting the jumps rates and the probability of failure on demand to zero.

Therefore, for each cluster, we propose to use an additional replicate to generate $\mathbf{a}_{\tau_{k+1}}^{k,j}$ and compute exactly its contribution. So for any $j \in \{1, \dots, N_k\}$, we will extend the selected trajectory $\mathbf{Z}_{\tau_k}^j$, N_k^j times, where $N_k^j = \tilde{N}_k^j + \mathbb{1}_{\tilde{N}_k^j > 0}$. We denote j_i the index of the i^{th} replicate of $\mathbf{Z}_{\tau_k}^j$, and consider the added replicate has index 0 such that for $i \in \{0, \dots, \tilde{N}_k^j\}$ we have $\tilde{\mathbf{Z}}_{\tau_k}^{j_i} = \mathbf{Z}_{\tau_k}^j$. The additional replicate is deterministically extended to the preponderant trajectory, so we have $\mathbf{Z}_{\tau_{k+1}}^{j_0} = \mathbf{a}_{\tau_{k+1}}^{k,j}$, and we set its weight to $W_{k+1}^{j_0} = \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) \frac{\tilde{N}_k^j}{N_k}$, so that it carries all the mass associated to the preponderant trajectories of a cluster. Then we can use all the remaining \tilde{N}_k^j trajectories in the cluster to estimate the non preponderant part of the cluster's propagation (the 1st term in the right hand side of equation (11.3)). For $i > 0$, we condition the extensions to avoid $\mathbf{a}_{\tau_{k+1}}^{k,j}$ generating the $\mathbf{Z}_{\tau_{k+1}}^{j_i}$ according to the kernel $\bar{\mathbf{V}}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$ and set $W_{k+1}^{j_i} = \frac{1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)}{\tilde{N}_k^j} \frac{\tilde{N}_k^j}{N_k}$. Usually, the simulations of a restricted law are carried out using a rejection algorithm, but in our case a rejection algorithm would perform poorly. The rate of rejection would be too high, as it would be equal to $\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)$ which is typically close to 1. For PDMPs, such simulations, conditioned to avoid a preponderant trajectory, can be efficiently carried out using the memorization method. This method, introduced in [37], shares similarities with the inverse method. It therefore benefits from not using any rejection, and so it is well suited to our applications. The memorization method is presented in chapter 10.

The target distributions $\tilde{\eta}_k$ are still estimated with $\tilde{\eta}_k^{\tilde{N}_k}$, using equation (9.31), but for $k = 0$ to $n - 1$, η_{k+1} , the propagation of a target distribution, is now estimated by :

$$\begin{aligned} \eta_{k+1}^{N_{k+1}} &= \sum_{i=1}^{N_{k+1}} W_{k+1}^i \delta_{\mathbf{Z}_{\tau_{k+1}}^i} = \sum_{j=1, \tilde{N}_k^j > 0}^{N_k} \sum_{i=0}^{\tilde{N}_k^j} W_{k+1}^{j_i} \delta_{\mathbf{Z}_{\tau_{k+1}}^{j_i}} \\ &= \sum_{j=1, \tilde{N}_k^j > 0}^{N_k} \frac{\tilde{N}_k^j}{N_k} \left[\mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) \delta_{\mathbf{a}_{\tau_{k+1}}^{k,j}} + \frac{(1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j))}{\tilde{N}_k^j} \sum_{i=1}^{\tilde{N}_k^j} \delta_{\mathbf{Z}_{\tau_{k+1}}^{j_i}} \right] \end{aligned} \quad (11.4)$$

Let $\tilde{\mathbf{N}}_k = (N_0, \tilde{N}_0, N_1, \tilde{N}_1, \dots, \tilde{N}_k)$ and $\mathbf{N}_k = (N_0, \tilde{N}_0, N_1, \tilde{N}_1, \dots, N_k)$. We now note $\tilde{\gamma}_k^{\tilde{\mathbf{N}}_k}$ and $\gamma_k^{\mathbf{N}_k}$ the estimations of the unnormalized distributions, and for all $k \leq n$ and $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$, we define them by:

$$\tilde{\gamma}_k^{\tilde{\mathbf{N}}_k}(f) = \tilde{\eta}_k^{\tilde{\mathbf{N}}_k}(f) \prod_{s=0}^{k-1} \eta_s^{N_s}(G_s) \quad \text{and} \quad \gamma_k^{\mathbf{N}_k}(f) = \eta_k^{N_k}(f) \prod_{s=0}^{k-1} \eta_s^{N_s}(G_s). \quad (11.5)$$

In the end, p is estimated using the equation :

$$\hat{p} = \eta_n^{N_n}(f_h) \prod_{k=0}^{n-1} \eta_k^{N_k}(G_k). \quad (11.6)$$

The full modified version of the algorithm is presented in Figure 11.1. We call this modified version of the IPS algorithm the IPS+M algorithm.

Throughout the rest of the manuscript, the notation \mathbb{E}_M will indicate that the expectation is associated to the IPS+M method and \mathbb{E} will still denote the expectation for the original IPS method.

Initialization : $k = 0$, $\forall j = 1..N$, $\mathbf{Z}_0^j = (z_0)$ and $W_0^j = \frac{1}{N}$, and $\tilde{W}_0^j = \frac{G_0(\mathbf{Z}_0^j)}{\sum_s G_0(\mathbf{Z}_0^s)}$

while $k < n$ **do**

Selection:

set $\tilde{N}_k = N$, and sample $(\tilde{N}_k^j)_{j=1..N_k} \sim \text{Mult}(\tilde{N}_k, (\tilde{W}_k^j)_{j=1..N_k})$

$\forall i = 1..\tilde{N}_k$, $\tilde{W}_k^i := \frac{1}{\tilde{N}_k}$

$\forall j = 1..N_k$, set $N_k^j = \tilde{N}_k^j + \mathbb{1}_{\tilde{N}_k^j > 0}$

set $N_{k+1} := \sum_{j=1}^{N_k} N_k^j$

Propagation :

for $j := 1..\tilde{N}_k$ **do**

if $N_k^j > 0$ **then**

set $\mathbf{Z}_{\tau_{k+1}}^{j_0} = \mathbf{a}_{\tau_{k+1}}^{k,j}$ and $W_{k+1}^{j_0} = \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j) \sum_{i=1}^{\tilde{N}_k^j} \tilde{W}_k^{j_i}$

for $j = 1..\tilde{N}_k^j$ **do**

$\mathbf{Z}_{\tau_{k+1}}^{j_i} \sim \bar{\mathbf{V}}_{k+1}(\cdot | \mathbf{Z}_{\tau_k}^j)$ and set $W_{k+1}^{j_i} = \frac{(1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j))}{\tilde{N}_k^j} \sum_{i=1}^{\tilde{N}_k^j} \tilde{W}_k^{j_i}$

for $i := 1..N_{k+1}$ **do**

$\tilde{W}_{k+1}^i = \frac{W_{k+1}^i G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^i)}{\sum_j W_{k+1}^j G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^j)}$

if $\forall j$, $\tilde{W}_{k+1}^j = 0$ **then**

$\forall q > k$, set $\eta_q^{N_q} = \tilde{\eta}_q^{\tilde{N}_q} = 0$ and Stop

else

$k := k + 1$

Figure 11.1 – IPS+M algorithm

11.2 Convergence properties of the IPS+M estimators

In this section we show that the estimator \hat{p} of the IPS+M method has the same basic properties as the IPS estimator. With the IPS+M method, \hat{p} converges almost surely to p , it is unbiased, and it satisfies a CLT. The proofs that we provide in this section follow the reasoning of the proofs in [21]. We present how to adjust the original proofs to take into account that the extensions of the trajectories within a cluster are no longer identically distributed. Finally we show that the asymptotic variance of the CLT is reduced with the IPS+M method.

The martingale decomposition of the anticipated biases

For $p \leq 2n$, we define \mathcal{F}_p the filtration associated to the sequence of the p first random samples built with the IPS+M algorithm: $((\mathbf{Z}_{\tau_0}^j)_{j \leq N_0}, (\tilde{\mathbf{Z}}_{\tau_0}^j)_{j \leq N}, (\mathbf{Z}_{\tau_1}^j)_{j \leq N_1}, \dots)$. So when p is an even number such that $p = 2k$, \mathcal{F}_p is the filtration generated by the vector $((\mathbf{Z}_{\tau_0}^j)_{j \leq N_0}, (\tilde{\mathbf{Z}}_{\tau_0}^j)_{j \leq N}, \dots, (\mathbf{Z}_{\tau_k}^j)_{j \leq N_k}, (\tilde{\mathbf{Z}}_{\tau_k}^j)_{j \leq N})$. For an odd number $p = 2k + 1$, \mathcal{F}_p is the filtration generated by the vector $((\mathbf{Z}_{\tau_0}^j)_{j \leq N_0}, (\tilde{\mathbf{Z}}_{\tau_0}^j)_{j \leq N}, \dots, (\mathbf{Z}_{\tau_k}^j)_{j \leq N_k})$. For $f \in \mathcal{M}(\mathbf{E}_{\tau_n})$ we let $\Gamma_{p,2n}^{\mathbf{N}}(h)$ be defined by

$$\begin{aligned} \Gamma_{2k,2n}^{\mathbf{N}}(f) &= \gamma_k^{\mathbf{N}^k}(Q_{k,n}(f)) - \gamma_k(Q_{k,n}(f)) \\ &= \gamma_k^{\mathbf{N}^k}(Q_{k,n}(f)) - \gamma_n(f) \end{aligned} \quad (11.7)$$

and

$$\begin{aligned} \Gamma_{2k+1,2n}^{\mathbf{N}}(f) &= \tilde{\gamma}_k^{\tilde{\mathbf{N}}^k}(\mathbf{V}_{k+1}Q_{k+1,n}(f)) - \tilde{\gamma}_k(\mathbf{V}_{k+1}Q_{k+1,n}(f)) \\ &= \tilde{\gamma}_k^{\tilde{\mathbf{N}}^k}(\mathbf{V}_{k+1}Q_{k+1,n}(f)) - \gamma_n(f). \end{aligned} \quad (11.8)$$

Using a telescopic argument we get

$$\begin{aligned} \Gamma_{p,2n}^{\mathbf{N}}(f) &= \sum_{k=0}^{2\lfloor \frac{p}{2} \rfloor} \gamma_k^{\mathbf{N}^k}(Q_{k,n}(f)) - \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}^{k-1}}(\mathbf{V}_k Q_{k,n}(f)) \\ &\quad + \mathbb{1}_{p>0} \sum_{k=1}^{2\lfloor \frac{p+1}{2} \rfloor} \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}^{k-1}}(\mathbf{V}_k Q_{k,n}(f)) - \gamma_{k-1}^{\mathbf{N}^{k-1}}(Q_{k-1,n}(f)), \end{aligned} \quad (11.9)$$

with the convention for $k = 0$, $\tilde{\gamma}_{-1}^{\tilde{N}_{-1}}(\mathbf{V}_{-1}Q_{0,n}(f)) = \gamma_n(f)$.

Noticing that $\gamma_k^{\mathbf{N}_k}(1) = \tilde{\gamma}_{k-1}^{\tilde{N}_{k-1}}(1) = \gamma_{k-1}^{\mathbf{N}_{k-1}}(G_{k-1})$, we can rewrite (11.9) as

$$\begin{aligned} \Gamma_{p,2n}^{\mathbf{N}}(f) &= \sum_{k=0}^{\lfloor \frac{2p}{2} \rfloor} \gamma_k^{\mathbf{N}_k}(1) \left(\eta_k^{\mathbf{N}_k}(Q_{k,n}(f)) - \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}} \mathbf{V}_{k-1}(Q_{k,n}(f)) \right) \\ &\quad + \mathbb{1}_{p>0} \sum_{k=1}^{\lfloor \frac{2p+1}{2} \rfloor} \tilde{\gamma}_{k-1}^{\tilde{N}_{k-1}}(1) \left(\tilde{\eta}_{k-1}^{\tilde{N}_{k-1}}(\mathbf{V}_{k-1}Q_{k,n}(f)) - \Psi_{k-1}(\eta_{k-1}^{\mathbf{N}_{k-1}}(\mathbf{V}_k Q_{k,n}(f))) \right), \end{aligned} \quad (11.10)$$

where for $k = 0$, we use the convention $\gamma_0^{\mathbf{N}_0}(1)\tilde{\eta}_{-1}^{\tilde{N}_{-1}}(\mathbf{V}_0Q_{0,n}(f)) = \gamma_n(f)$. The benefit of this decomposition is that it distinguishes the errors associated to the propagation steps and the errors associated to the selection steps. For the propagation steps, using (11.1) we easily get that for any $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$:

$$\mathbb{E}_M \left[\eta_k^{\mathbf{N}_k}(f) \mid \mathcal{F}_{2k-1} \right] = \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}} \mathbf{V}_k(f). \quad (11.11)$$

For the selection steps, as the resampling schemes are the same ones as for the IPS algorithm, we still have for any $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$:

$$\mathbb{E}_M \left[\tilde{\eta}_k^{\tilde{N}_k}(f) \mid \mathcal{F}_{2k} \right] = \Psi_k(\eta_k^{\mathbf{N}_k}(f)). \quad (11.12)$$

Thus, each selection step and propagation step is conditionally unbiased. Note that $\gamma_k^{\mathbf{N}_k}(1)$ is \mathcal{F}_{2k-1} -measurable and $\tilde{\gamma}_k^{\tilde{N}_k}(1)$ is \mathcal{F}_{2k} -measurable, so, when the samples are generated with the IPS+M algorithm, $(\Gamma_{p,2n}^{\mathbf{N}}(h))_{p \leq 2n}$ is a \mathcal{F}_p -martingale. Therefore, \hat{p} stays unbiased with the IPS+M method, because

$$\mathbb{E}_M[\Gamma_{2n,2n}^{\mathbf{N}}(f_h)] = \mathbb{E}_M[\hat{p} - p] = 0.$$

Almost sure convergence

Thanks to this martingale decomposition, we can use the same arguments as in the proof in Chapter 7.4.2 in [21]. Therefore the Theorems 7.4.2 and 7.4.3 page 239 and 241 are satisfied with the IPS+M method too, which yields the following theorem:

Theorem 27. *For any $h \in \mathcal{M}(\mathbf{E}_{\tau_n})$, \hat{p} converges almost surely to p , and, for any $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$, $\eta_k^{\mathbf{N}_k}(f)$ converges to $\eta_k(f)$ almost surely, $\tilde{\eta}_k^{\tilde{N}_k}(f)$ converges to $\tilde{\eta}_k(f)$ almost surely.*

A Central Limit Theorem

Theorem 28. *If the potential functions satisfy the assumption (G-PDMP) (page 128) and the samples are generated with the IPS+M algorithm, then we have the following*

convergence in distribution:

$$\sqrt{N}(\hat{p} - p) \xrightarrow{N \rightarrow \infty} \mathcal{N}(0, \sigma_{M,G}^2),$$

where

$$\begin{aligned} \sigma_{M,G}^2 &= \eta_0 \left(\left[Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left(\left(1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{\mathbf{V}}_k \left[Q_{k,n}(f_h) - \bar{\mathbf{V}}_k Q_{k,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \tilde{\gamma}_k(1)^2 \tilde{\eta}_{k-1} \left(\left[\mathbf{V}_k Q_{k,n}(f_h) - \tilde{\eta}_{k-1} \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right). \end{aligned} \quad (11.13)$$

Proof. Similarly to what is done in [21], in order to prove that \hat{p} satisfies a CLT, we begin by proving that the errors associated to the selection and propagation steps are normally distributed using Lindeberg's theorem.

For a sequence of function $(f_k)_{k \leq 2n}$ such that f_{2k} and f_{2k+1} are in $\mathcal{M}(\mathbf{E}_{\tau_k})$, we define the sum of errors until the p^{th} selection and propagation by:

$$\begin{aligned} M_{p,2n}^N(f) &= \sum_{k=0}^{\lfloor \frac{p}{2} \rfloor} \eta_k^{N_k}(f_{2k}) - \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}} \mathbf{V}_k(f_{2k}) \\ &+ \mathbb{1}_{p>0} \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}}(f_{2k-1}) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}})(f_{2k-1}). \end{aligned} \quad (11.14)$$

For $j \in \{1, \dots, N\}$ we let

$$U_{(2k+1)N+j}^N(f) = \frac{1}{\sqrt{N}} \left(f_{2k+1}(\tilde{\mathbf{Z}}_{\tau_k}^j) - \Psi_k(\eta_k^{N_k})(f_{2k+1}) \right). \quad (11.15)$$

For $k \geq 0$, $j \in \{1, \dots, N_k\}$ and $i \in \{0, \dots, N^j\}$, we consider that the indices j_i are ordered in such way that $j_0 > N$ and $j_i < N$ when $i > 0$. With such indexing $\forall s \in \{1, \dots, N\}$, $\exists j \in \{1, \dots, N_k\}$ and $i \in \{1, \dots, N^j\}$ such that $s = j_i$, and for such s we let

$$U_{2kN+s}^N(f) = \frac{1 - \mathbf{V}_{k+1}(\{\mathbf{a}_{\tau_{k+1}}^{k,j}\} | \mathbf{Z}_{\tau_k}^j)}{\sqrt{N}} \left(f_{2(k+1)}(\mathbf{Z}_{\tau_{k+1}}^{j_i}) - \bar{\mathbf{V}}_{k+1}(f_{2(k+1)})(\mathbf{Z}_{\tau_k}^j) \right). \quad (11.16)$$

For $j \in \{1, \dots, N_0\}$, let

$$U_j^N(f) = \frac{1}{\sqrt{N}} \left(f_0(\mathbf{Z}_0^j) - \eta_0(f_0) \right). \quad (11.17)$$

Thus,

$$\sqrt{N} M_{p,2n}^N(f) = \sum_{k=0}^{(p+1)N} U_k^N(f). \quad (11.18)$$

Noting \mathcal{P}_k^N a filtration adapted to the k first trajectories generated in the IPS+M algorithm. Note that we have that $\mathbb{E} [U_k^N(f) | \mathcal{P}_{k-1}^N] = 0$, and $\mathbb{E} [U_k^N(f)^2 | \mathcal{P}_{k-1}^N] < \infty$, and $|U_k^N(f)| < \frac{2}{\sqrt{N}} \sup_{k \leq n, \mathbf{Z}_{\tau_k} \in \mathbf{E}_{\tau_k}} \{|f_{2k}(\mathbf{Z}_{\tau_k})| \wedge |f_{2k+1}(\mathbf{Z}_{\tau_k})|\}$, so the Lindeberg condition is clearly satisfied. Then, we have that

$$\begin{aligned}
 \langle \sqrt{N} M_{p,2n}^{\mathbf{N}}(f) \rangle_p &= \sum_{k=0}^{(p+1)N} \mathbb{E} [U_k^N(f)^2 | \mathcal{P}_{k-1}^N] \\
 &= \eta_0^N \left([f_0 - \eta_0^N(f_0)]^2 \right) \\
 &\quad + \sum_{k=1}^{\lfloor \frac{p}{2} \rfloor} \tilde{\eta}_{k-1}^N \left(\left(1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{\mathbf{V}}_k [f_{2k} - \bar{\mathbf{V}}_k f_{2k}]^2 \right) \\
 &\quad + \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\eta}_{k-1}^N \left([f_{2k-1}(\mathbf{Z}_{\tau_{k-1}}) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}}) f_{2k-1}]^2 \right).
 \end{aligned} \tag{11.19}$$

As $\eta_k^{N_k}$ and $\tilde{\eta}_k^N$ converge almost surely to η_k and $\tilde{\eta}_k$, $\langle \sqrt{N} M_{p,2n}^{\mathbf{N}}(f) \rangle_n$ converge in probability to

$$\begin{aligned}
 \sigma_p^2(f) &= \eta_0 \left([f_0 - \eta_0(f_0)]^2 \right) \\
 &\quad + \sum_{k=1}^{\lfloor \frac{p}{2} \rfloor} \tilde{\eta}_{k-1} \left(\left(1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{\mathbf{V}}_k [f_{2k} - \bar{\mathbf{V}}_k f_{2k}]^2 \right) \\
 &\quad + \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\eta}_{k-1} \left([f_{2k-1}(\mathbf{Z}_{\tau_{k-1}}) - \tilde{\eta}_{k-1} f_{2k-1}]^2 \right).
 \end{aligned} \tag{11.20}$$

By application of the Lindeberg's theorem for triangular array (see for instance Theorem 4 on page 543 in [49]), we get that $\sqrt{N} M_{p,2n}^{\mathbf{N}}(f)$ converges in law to a centered Gaussian of variance $\sigma_p^2(f)$. As a corollary, if for $p \neq 2k$ we take $f_p = 0$ and for $p = 2k$ $f_{2k} = Q_{k,n}(f_h)$, we get that

$$\begin{aligned}
 &\sqrt{N} \left(\eta_k^{N_k} Q_{k,n}(f_h) - \tilde{\eta}_{k-1}^N \mathbf{V}_k Q_{k,n}(f_h) \right) \\
 &\quad \xrightarrow{N \rightarrow \infty} \tilde{\eta}_{k-1} \left(\left(1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{\mathbf{V}}_k [Q_{k,n}(f_h) - \bar{\mathbf{V}}_k Q_{k,n}(f_h)]^2 \right)
 \end{aligned}$$

and if for $p \neq 2k - 1$ we take $f_p = 0$ and for $p = 2k - 1$ $f_{2k-1} = \mathbf{V}_k Q_{k,n}(f_h)$, we get that

$$\begin{aligned} \sqrt{N} \left(\tilde{\eta}_{k-1}^N (\mathbf{V}_k Q_{k,n}(f_h)) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}}) (\mathbf{V}_k Q_{k,n}(f_h)) \right) \\ \xrightarrow{N \rightarrow \infty} \tilde{\eta}_{k-1} \left(\left[\mathbf{V}_k Q_{k,n}(f_h) - \tilde{\eta}_{k-1} \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right). \end{aligned}$$

As $\gamma_k^{N_k}(1)^2$ and $\tilde{\gamma}_k^N(1)^2$ converges almost surely to $\gamma_k(1)^2$ and $\tilde{\gamma}_k(1)^2$, by an application of Slutsky's Lemma, we get that $\sqrt{N} \Gamma_{2N,2n}^N(f_h)$ converges in law to a centered Gaussian with variance

$$\begin{aligned} \sigma_{M,G}^2 &= \gamma_0(1)^2 \eta_0 \left(\left[Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left(\left(1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{\mathbf{V}}_k \left[Q_{k,n}(f_h) - \bar{\mathbf{V}}_k Q_{k,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \tilde{\gamma}_{k-1}(1)^2 \tilde{\eta}_{k-1} \left(\left[\mathbf{V}_k Q_{k,n}(f_h) - \tilde{\eta}_{k-1} \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right). \end{aligned} \quad (11.21)$$

□

Variance reduction

Theorem 29. *The variance of the original IPS can be decomposed as follows:*

$$\sigma_{IPS,G}^2 = \sigma_{M,G}^2 + \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left(v_k(\mathbf{Z}_{\tau_{k-1}}) \bar{\mathbf{V}}_k \left(\left[Q_{k,n}(f_h)(\mathbf{a}_{\tau_k}) - Q_{k,n}(f_h)(\mathbf{Z}_{\tau_k}) \right]^2 \right) \right), \quad (11.22)$$

where $v_k(\mathbf{Z}_{\tau_{k-1}}) = \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}) (1 - \mathbf{V}_k(\{\mathbf{a}_{\tau_k}\} | \mathbf{Z}_{\tau_{k-1}}))$. Therefore we have $\sigma_{M,G}^2 \leq \sigma_{IPS,G}^2$.

Proof.

$$\sigma_{IPS,G}^2 = \sum_{k=0}^n \gamma_k(1)^2 \eta_k \left(\left[Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h) \right]^2 \right) \quad (11.23)$$

$$\begin{aligned} &= \eta_0 \left(\left[Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \mathbf{V}_k \left(\left[Q_{k,n}(f_h) - \mathbf{V}_k Q_{k,n}(f_h) + \mathbf{V}_k Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h) \right]^2 \right) \\ &= \eta_0 \left(\left[Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \mathbf{V}_k \left(\left[Q_{k,n}(f_h) - \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \tilde{\gamma}_{k-1}(1)^2 \tilde{\eta}_{k-1} \left(\left[\mathbf{V}_k Q_{k,n}(f_h) - \tilde{\eta}_{k-1} \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right) \end{aligned} \quad (11.24)$$

Temporarily using the notation $\mathbf{V}_k(\{\mathbf{a}_{\tau_k}\}|\mathbf{Z}_{\tau_{k-1}}) = p_k$, for any $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$, we get

$$\begin{aligned}
 & \mathbf{V}_k \left(\left[f(\mathbf{Z}_{\tau_k}) - \mathbf{V}_k f \right]^2 \right) \\
 &= \mathbf{V}_k \left(\left[f(\mathbf{Z}_{\tau_k}) - p_k f(\mathbf{a}_{\tau_k}) - (1-p_k) \bar{\mathbf{V}}_k(f) \right]^2 \right) \\
 &= \mathbf{V}_k \left(f(\mathbf{Z}_{\tau_k})^2 - 2p_k f(\mathbf{Z}_{\tau_k}) f(\mathbf{a}_{\tau_k}) - 2(1-p_k) f(\mathbf{Z}_{\tau_k}) \bar{\mathbf{V}}_k f \right. \\
 &\quad \left. + p_k^2 f(\mathbf{a}_{\tau_k})^2 + 2p_k f(\mathbf{a}_{\tau_k}) \bar{\mathbf{V}}_k f + (1-p_k)^2 (\bar{\mathbf{V}}_k f)^2 \right) \\
 &= p_k f(\mathbf{a}_{\tau_k})^2 + (1-p_k) \bar{\mathbf{V}}_k(f^2) \\
 &\quad - 2p_k^2 f(\mathbf{a}_{\tau_k})^2 - 2p_k(1-p_k) f(\mathbf{a}_{\tau_k}) \bar{\mathbf{V}}_k(f) \\
 &\quad - 2p_k(1-p_k) f(\mathbf{a}_{\tau_k}) \bar{\mathbf{V}}_k f - 2(1-p_k) f^2 \bar{\mathbf{V}}_k(f)^2 \\
 &\quad + p_k^2 f(\mathbf{a}_{\tau_k})^2 + 2p_k f(\mathbf{a}_{\tau_k}) \bar{\mathbf{V}}_k f + (1-p_k)^2 \bar{\mathbf{V}}_k(f)^2 \\
 &= p_k(1-p_k) \left[f(\mathbf{a}_{\tau_k})^2 - 2f(\mathbf{a}_{\tau_k}) \bar{\mathbf{V}}_k f + \bar{\mathbf{V}}_k(f^2) \right] \\
 &\quad + (1-p_k)^2 \left(\bar{\mathbf{V}}_k(f^2) - \bar{\mathbf{V}}_k(f)^2 \right) \\
 &= p_k(1-p_k) \bar{\mathbf{V}}_k \left(\left[f(\mathbf{a}_{\tau_k}) - f(\mathbf{Z}_{\tau_k}) \right]^2 \right) + (1-p_k)^2 \left(\bar{\mathbf{V}}_k(f^2) - \bar{\mathbf{V}}_k(f)^2 \right). \tag{11.25}
 \end{aligned}$$

In particular, for $f = Q_{k,n}(f_h)$ we get

$$\begin{aligned}
 & \mathbf{V}_k \left(\left[Q_{k,n}(f_h) - \mathbf{V}_k Q_{k,n}(f_h) \right]^2 \right) \\
 &= p_k(1-p_k) \bar{\mathbf{V}}_k \left(\left[Q_{k,n}(f_h)(\mathbf{a}_{\tau_k}) - Q_{k,n}(f_h) \right]^2 \right) + (1-p_k)^2 \left(\bar{\mathbf{V}}_k(f^2) - \bar{\mathbf{V}}_k(f)^2 \right) \tag{11.26}
 \end{aligned}$$

Plugging (11.26) into the second line of (11.24), yields (11.22). \square

Application to two test systems

12.1 Empirical confirmation

In order to confirm our results empirically, we have applied the IPS method and the IPS+M method to two two-components systems.

12.1.1 The Heated-room 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.

X_t represents the temperature of the room at time t . M_t represents the status of the heaters at time t . Heaters can be on, off, or out-of-order, so $\mathbb{M} = \{ON, OFF, F\}^2$. The state of the system is $Z_t = (X_t, M_t)$.

The differential equation that governs the temperature can be derived from the physics. x_e is the exterior temperature. β_1 is the rate of the heat transition with the exterior. β_2 is the heating power of each heater. The differential equation giving the evolution of the temperature of the room has the following form:

$$\frac{dX_t}{dt} = \beta_1(x_e - X_t) + \beta_2 \mathbb{1}_{M_t^1 \text{ or } M_t^2 = ON}.$$

The heaters are programmed to maintain the temperature within an interval (x_{min}, x_{max}) where $x_e < 0 < x_{min}$. We consider that the two heaters are in passive redundancy in the sense that: when $X \leq x_{min}$ the second heater activates only if the first one is failed. When a repair of a heater occurs, if $X \leq x_{min}$ and the other heater is failed, then the heater status is set to *ON*, else the heater status is set to *OFF*. To handle the programming of the heaters, we set $\Omega_m = (-\infty, x_{max})$ when all the heaters are failed $m = (F, F)$ or when at least one is activated, otherwise we set $\Omega_m = (x_{min}, x_{max})$.

Due to the continuity of the temperature, the reference measure for the Kernel is $\forall B \in \mathcal{B}(E)$, $\nu_{(x,m)}(B) = \sum_{m^+ \in \mathbb{M} \setminus \{m\}} \delta_{(x,m^+)}(B)$. On the top boundary in x_{max} , heaters turn off with probability 1. On the bottom boundary in x_{min} , when a heater is supposed

to turn on, there is a probability $\gamma = 0.01$ that the heater will fail on demand. So, for instance, if $z^- = (x_{min}, (OFF, OFF))$, we have $K_{z^-}(x_{min}, (ON, OFF)) = 1 - \gamma$, and $K_{z^-}(x_{min}, (F, ON)) = \gamma(1 - \gamma)$, and $K_{z^-}(x_{min}, (F, F)) = \gamma^2$.

Let j be a transition from m to m^+ . For the spontaneous jumps (jumps that does not happen on a boundary), if the transition j corresponds to the failure of a heater, then: $\lambda^j(x, m) = 0.0021 + 0.00015 \times x$ and, if the transition corresponds to a repair, then $\lambda^j(x, m) = 0.2$ when $M^j = F$. Here the system failure occurs when the temperature of the room falls below zero, so $D = \{(x, m) \in E, x < 0\}$. A possible trajectory of the state of this system is plotted in figure 1.17. The probability of failure p was estimated to 2.71×10^{-5} thanks to a massive Monte-Carlo of 10^7 simulations.

		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}

Table 12.1 – Empirical means and empirical variances on 100 runs with $N = 10^5$ for the MC, the IPS and the IPS+M methods

12.1.2 Results of the simulations on the Heated-room system

The results of the simulation study for the heated-room system are displayed in table 12.1. Here we have used the potential functions:

$$\forall k < n, \quad G(\mathbf{Z}_{\tau_k}) = \mathbb{1}_{\tau_D > \tau_k} \exp \left[-\alpha_1 (b(Z_{\tau_k}) + 1)^2 \right] + \mathbb{1}_{\tau_D \leq \tau_k}, \quad (12.1)$$

where $b(Z_{\tau_k})$ is the number of working components and τ_D is the time of the failure of the system. The value of α was set to 1.1. We have tried different values of α between 0.5 and 1.5 with a step 0.1. The value of $\alpha = 1.1$ was chosen among our trials as the one yielding the best variance reduction for the IPS method. For the IPS, IPS+M, and MC methods the variances are estimated empirically: we run the methods 100 times and we take the empirical variances of the 100 estimates. The results highlight that the IPS method is ill-suited to PDMPs, as it yields a higher variance than the MC method. Conversely, our IPS+M method performs well and has overcome the issue of the PDMP. Indeed, in the case $n = 10$, it reduces the variance by a factor 2.7 compared to the MC method, and by a factor 10 compared to the IPS method.

The IPS+M is about 4 times slower than the IPS method, so, in terms of computational

cost, the method is only 2.5 more efficient than the IPS method on this test case. For a run of $N = 10^5$ the IPS+M is about 2.7 time slower than Monte-carlo method. So in terms of computational cost the IPS+M is slightly more efficient than the Monte-Carlo Method.

12.1.3 Remark on the SMC with Memorization

The SMC method can be seen as an improvement of the IPS method. Therefore, one may think that the IPS+M algorithm can be improved in the form of a SMC+M algorithm by adding adaptive optional re-sampling steps to the IPS+M. However, it is not beneficial to add these adaptive optional re-sampling steps. Indeed we noticed that, as we greatly modify the propagation process, the weights are greatly imbalanced and the effective-sample-size ends up being extremely small, which would trigger the re-sampling each time. Therefore adding adaptive optional re-sampling to the IPS+M has no effect, and in practice the IPS+M methods and the SMC+M methods are the same.

12.1.4 A dam system

The second system models a dam subjected to an incoming water flow. The physical variable of interest is the water level in the dam denoted by X_t . The failure of the system occurs when the water level exceeds a security threshold $x_{lim} = 10$ before time $t_f = 50$. The initial level is set to $X_0 = 0$. The water flow is characterized by the input debit $Q = 10$. The dam has two evacuation valves with output debit Q . Each valve can be either open, close or stuck closed. So $\mathbb{M} = \{Open, Closed, Stuckclosed\}^2$. The valves are programmed in passive redundancy, so if the valves are in functioning order there is always one valve open and one valve closed. Though, the valve can get stuck closed and this happens at random times with exponential distribution with intensity $\lambda = 0.001$. The valves are repaired with a repair rate $\mu = 0.1$. When both valves are stuck closed the reservoir of the dam starts filling up according to the equation $\frac{dX_t}{dt} = Q/S$, where $S = 10$ is the surface of the reservoir.

12.1.5 Results of the simulations for the dam system

The results of the simulation study for the dam system are displayed in table 12.2. Here we have used the potential functions:

$$\forall k < n, \quad G(\mathbf{Z}_{\tau_k}) = \mathbb{1}_{\tau_D > \tau_k} \exp \left[\alpha_1 (x_{lim} - X_{\tau_k}) + \alpha_2 (b(Z_{\tau_k}) + 1)^2 \right] + \mathbb{1}_{\tau_D \leq \tau_k}, \quad (12.2)$$

where $b(Z_{\tau_k})$ is the number of working components. The value of α_1 was set to -0.9 and the value of α_2 was set to -1 (these are a priori guesses, we have not tried to use any optimization). For the IPS and the IPS+M methods the variances are estimated

empirically: we run the methods 50 times and we take the empirical variances of the 50 estimates. The results are presented in table 12.2. The results highlight that the IPS

		MC	IPS	IPS+M
	\hat{p}	1.12×10^{-4}		
	$\hat{\sigma}^2$	1.12×10^{-9}		
$n = 5$	\hat{p}		1.75×10^{-4}	1.12×10^{-4}
	$\hat{\sigma}^2$		3.08×10^{-8}	4.37×10^{-9}

Table 12.2 – Empirical means and empirical variances on 50 runs with $N = 10^5$ for the MC, the IPS and the IPS+M methods

method is again ill-suited to PDMPs, as it yields a variance 30 times larger than the MC method. Our IPS+M method performs better than the IPS method as the variance is reduced by a factor 7. Yet on this example the IPS+M method has not overcome the issue of the PDMP, as its variance is 3.4 times larger than the variance of the Monte-carlo estimator. In terms of computational cost, on this example the IPS+M method was 3.6 times slower than the IPS, and 11.8 times slower than the Monte-Carlo method. So the efficiency of the IPS+M is about 40 lower than the Monte-Carlo method. Clearly, the implementation of the IPS+M method requires a careful choice of the form of the potential functions and of their parameters.

Conclusion on the IPS+M

As the IPS method does not perform well when it is used on a concentrated PDMP, we introduce and analyze the IPS+M method, that is a modified version of the IPS that performs better with concentrated PDMP. The IPS+M method is similar to the IPS but has different propagation steps. Its propagation steps focus on clusters of identical particles rather than on particles individually. For each cluster a memorization method is used to get an empirical approximation of the distribution of the propagated cluster, which allows to greatly improve the accuracy of the method. We have shown that the proposed algorithm yields a strongly consistent estimation, and that this estimation satisfies a CLT. We prove that the asymptotic variance of the IPS+M estimator is always smaller than the asymptotic variance of the IPS estimator. Simulations also confirm these results, showing that the IPS+M can yield a variance reduction when the IPS cannot. In terms of computational cost, our implementations of the IPS+M method give approximately the same efficiency as the Monte-Carlo method in the examples considered in this paper, where the goal is to estimate a probability of the order of 10^{-5} for rather simple toy models. The numerical implementations certainly deserve more careful attention. We also believe that there are ways to improve the efficiency of the IPS+M method by finding a better class of potential functions. Another interesting improvement to the IPS+M method would be to propose an estimator of the variance. We believe that it should be possible to adapt one of the estimators proposed in [38] for the IPS method in order to get an estimator of the variance for the IPS+M estimator.

PART V

Conclusion and prospects for future work

Our goal is to accelerate the reliability assessment of power generation systems done within the EDF's toolbox PyCATSHOO. This reliability assessment is originally done by a Monte-Carlo method, and based on the stochastic model of Piecewise deterministic Markovian processes (PDMP)[19, 18]. Our work focuses on two methods used for Monte-Carlo acceleration : The importance sampling (IS) method [39, 47], and the interacting particles system (IPS) method [23, 21, 22]. These methods are designed to estimate an integral or a probability with a better accuracy than the Monte-Carlo method. We propose adaptations of these methods to the PDMP case in order to assess the reliability of power generation systems with a better accuracy and therefore with less simulation runs.

Conclusion on the IS method

The IS method is adapted to the PDMP case

Concerning the IS method, our work [15] makes the use of importance sampling on PDMPs with boundaries possible, and it provides a convenient way to specify the law of the importance process. We have developed the theoretical foundations of importance sampling for PDMPs with boundaries. The challenge was to properly define the likelihood ratio involved in the weighting of the simulation outputs when the random variable of interest is a trajectory of PDMP with boundaries. We have exhibited the reference measure ζ on the space of PDMP trajectories, which allowed us to identify the closed form expression of the density of a PDMP trajectory, and to properly define the likelihood ratio. Thanks to the reference measure, we have also been able to identify the possible importance processes. Then we have studied the characteristics of the optimal process to assess the quantity $\mathbb{E}[h(\mathbf{Z})]$ where $h \in \mathcal{M}(\mathbf{E}_{t_f})$. The distribution of this process has density $g^*(\mathbf{z}) = \frac{h(\mathbf{z})f(\mathbf{z})}{p}$, when used as an importance process it brings the variance of the IS estimator to zero. We have managed to show that the closed form expression of the law of this optimal process depends on a function U^* defined by:

$$U^* : \bigcup_{s \in [0, t_f]} \mathbf{E}_s \times \{s\} \longrightarrow \mathbb{R}$$

$$(\mathbf{z}, s) \longrightarrow \mathbb{E}[h(\mathbf{Z}) | \mathbf{Z}_s = \mathbf{z}]$$

and to the function $U^- = \int U^* d\mathcal{K}$. Indeed, we have shown that the density of the time between, for instance, the k -th jump and of the $k + 1$ -th jump is such that, $\forall s \in [0, t_f), \mathbf{z}_s \in \mathbf{E}_s, u \in [0, t_{z_s}^*)$:

$$g_{T'_k | \mathbf{Z}'_{S'_k} = \mathbf{z}_s, S'_k = s}(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s + u)}{U^*(\mathbf{z}_s, s)} f_{T_k | Z_{S_k} = z_s}(u).$$

We have shown that the transition kernels and the jump rates associated to this optimal process are such that, $\forall s \in [0, t_f)$, $\mathbf{z}_s \in \mathbf{E}_s$, $u \in [0, t_{z_s}^*)$:

$$K_{\mathbf{z}_s^-, s}^*(z) = \frac{U^*(\mathbf{z}_s, s)}{U^-(\mathbf{z}_s^-, s)} K_{z_s^-}(z_s), \quad (13.1)$$

$$\lambda_{\mathbf{z}_s, s}^*(u) = \frac{U^-(\Phi_{\mathbf{z}_s}(u), s + u)}{U^*(\Phi_{\mathbf{z}_s}(u), s + u)} \lambda_{z_s}(u). \quad (13.2)$$

As finding an efficient importance density is easier when we restrict the search to a family of parametric importance densities, we propose a way to specify the family of parametric importance densities. We have proposed to approximate the function U^* by parametric functions U_α , and to plug this approximation instead of U^* into the expressions (13.1) and (13.2) of the optimal transition kernel densities and jump rates.

Inefficiency on a big system

This strategy yields spectacular results on the small system of the heated room, which is very encouraging, but so far the IS method does not work with the second case of industrial size. In order to fit the parameters of the importance process, we used the Cross entropy method. We believe the inefficiency on the industrial system is due to two main factors. The first reason being that the system is complex and has many ways to fail. Indeed the parametric function U_α must give an importance density that favors all the ways to fail, but it must also favor a way to fail proportionally to its probability, which is difficult to do. The second reason is that the cross-entropy is not well suited to find the optimal parameters when the failure region is not connected and is made of many parts.

Prospect on IS

In order to improve the developed IS method for PDMP, we need to find a way to specify better U_α functions, and to improve the optimization of the parameter. We believe that the design of the U_α should depend on the identification of the minimal groups, and that the classic tools of reliability analysis can help to the identification of the minimal groups, and therefore can help the design of a family of U_α functions containing a function close to the optimal proximity function U^* . We also believe that in order to avoid over-biasing situations, the optimization method used (the cross entropy) should handle the fact that the optimal density may be, in some sense yet to be defined, multimodal, or the fact that the optimal density may concentrate its probability mass on distinct parts of the trajectory space.

More generally, for the use of IS method with any model, it is important to develop a tool to identify a situation with over-biasing and under-estimated variance.

Conclusion on the IPS method

Tuning of the IPS method

In part III, we provide the closed form expressions of the optimal potential functions $(G_k^*)_{k < n}$ for which the variance of the IPS estimator is minimized. This work is also presented in the preprint [16]. It is interesting to note that the expressions of the G_k 's depend on the function U^* also involved in the optimal importance process of the IS method. As the optimal potentials $(G_k^*)_{k < n}$ yield a positive variance, it confirms that the reductions of variance achievable with the IPS method is lower bounded. This result having already been observed empirically or analytically on some examples. The dependency of the G_k 's on U^* also confirms that the IS method is preferable to the IPS method when we are able to closely approximate U^* and when the importance process can be simulated.

Concerning the search of the best potential functions within a family of parametric potential functions, notice that the new possibility to estimate the variance of the estimators of the IPS [1] gives a new brick to build a method of optimization of the parameter. With these estimators, it might be possible to build an IPS algorithm that optimizes the parameters of the potentials, like the minimization of the variance method [48, pages 62,83] does for IS.

The IPS+M method is better suited to a concentrated PDMP case

In part IV, we develop the IPS+M method which is an adaption on the IPS method to the case of a concentrated PDMP. This work is also presented in the paper [14]. This method consists in modifying the propagation steps of the IPS method: it modifies the propagation of identical trajectories using the memorization method, and correct the induced bias by appropriately weighting the propagated particles. We prove that the IPS+M method yields a strongly consistent estimator that satisfies a CLT. We prove that the asymptotic variance of the estimator is always smaller with the IPS+M method than with the IPS method.

The unbalanced weights within the IPS+M methods, makes it impossible to use adaptive resampling with the effective sample size criteria.

A suggestion for improvement of the IPS+M

The benefit of the IPS+M method is that it forces the generation of trajectories with one jump between two consecutive subdivisions $[\tau_k, \tau_{k+1}]$. This allows to better explore the set of trajectories and yields a smaller variance of the estimator. But we can notice that the failing trajectories may very well jump more than once between two consecutive

subdivisions $[\tau_k, \tau_{k+1}]$, and that our IPS+M method does not force the exploration of such trajectories with more than one jump between two consecutive subdivisions $[\tau_k, \tau_{k+1}]$. We think that we could reach an even smaller variance if the method would also generate trajectories with several close jumps with a higher probability. We believe that one good way to generate trajectories with any length between two consecutive jumps is to no longer use a subdivision on fixed times $(\tau_k)_{k \leq n}$ but to use a subdivision on jumps, setting for each trajectory $\tau_k = S_k$. The issue with that option is that, at a the $k - th$ step of the algorithm, we get a sample of trajectories containing k jumps, but these trajectories are not of the same length. The challenge is then to find a way so that the last sample of trajectories (at the $n - th$ step) contains only trajectories of length t_f , so that we can estimate the quantity $p = \mathbb{P}(\mathbf{Z}_{t_f} \in \mathcal{D})$.

Optimal intensity's expression, and some properties of U^*

A.1 Proof of Equality (4.11)

Let \mathbf{z}_s be a trajectory of size s . Remember that equality (4.11) states that

$$U^*(\Phi_{\mathbf{z}_s}(t_{z_s}^*), s + t_{z_s}^*) = \lim_{t \nearrow t_{z_s}^*} U^*(\Phi_{\mathbf{z}_s}(t), s + t).$$

Proof. We denote by T the time until the next jump after the trajectory has reached the state $Z_{s+t} = \phi_{z_s}(t)$. Then we have:

$$\begin{aligned} U^*(\Phi_{\mathbf{z}_s}(t), s + t) &= \mathbb{E}[|h(\mathbf{z})| \mid \mathbf{Z}_{s+t} = \Phi_{\mathbf{z}_s}(t), s + t] \\ &= \mathbb{E}\left[\mathbb{E}[|h(\mathbf{z})| \mid \mathbf{Z}_{s+t+T}] \mid \mathbf{Z}_{s+t} = \Phi_{\mathbf{z}_s}(t)\right] \\ &= \mathbb{E}\left[(\mathbb{1}_{T < t_{\Phi_{z_s}(t)}^*} + \mathbb{1}_{T = t_{\Phi_{z_s}(t)}^*}) U^*(\mathbf{Z}_{s+t+T}) \mid \mathbf{Z}_{s+t} = \Phi_{\mathbf{z}_s}(t)\right] \\ &= \int_0^{t_{\Phi_{z_s}(t)}^*} U^-(\Phi_{\Phi_{z_s}(t)}(u), s + t + u) \lambda_{\Phi_{z_s}(t)}(u) \exp\left[-\Lambda_{\Phi_{z_s}(t)}(u)\right] du \\ &\quad + \exp\left[-\Lambda_{\Phi_{z_s}(t)}(t_{\Phi_{z_s}(t)}^*)\right] \int_E K_{z^-}(z_{s+t+t_{\Phi_{z_s}(t)}^*}) U^*(\mathbf{z}_{s+t+t_{\Phi_{z_s}(t)}^*}) d\nu_{z^-}(z_{s+t+t_{\Phi_{z_s}(t)}^*}) \end{aligned}$$

where $z^- = \Phi_{\Phi_{z_s}(t)}(t_{\Phi_{z_s}(t)}^*)$

$$\begin{aligned} U^*(\Phi_{\mathbf{z}_s}(t), s + t) &= \int_t^{t_{z_s}^*} U^-(\Phi_{\mathbf{z}_s}(u), s + u) \lambda_{z_s}(u) \exp\left[-\Lambda_{z_s}(u) + \Lambda_{z_s}(t)\right] du \\ &\quad + \exp\left[-\Lambda_{\Phi_{z_s}(t)}(t_{z_s}^* - t)\right] \int_E K_{z^-}(z_{s+t_{z_s}^*}) U^*(\mathbf{z}_{s+t_{z_s}^*}) d\nu_{z^-}(z_{s+t_{z_s}^*}) \end{aligned}$$

where $z^- = \Phi_{z_s}(t_{z_s}^*)$

so $U^*(\Phi_{\mathbf{z}_s}(t), s + t) = o(1) + (1 + o(1))U^-(\Phi_{\mathbf{z}_s}(t_{z_s}^*))$ as $t \rightarrow t_{z_s}^*$, $t < t_{z_s}^*$. \square

A.2 Proof of theorem 18

Proof. We have seen in the proof above that

$$\begin{aligned} U^*(\Phi_{\mathbf{z}_s}(t), s+t) &= \int_t^{t_{z_s}^*} U^-(\Phi_{\mathbf{z}_s}(u), s+u) \lambda_{z_s}(u) \exp[-\Lambda_{z_s}(u) + \Lambda_{z_s}(t)] du \\ &\quad + \exp[-\Lambda_{\Phi_{\mathbf{z}_s}(t)}(t_{z_s}^* - t)] \int_E K_{z^-}(z_{s+t_{z_s}^*}) U^*(\mathbf{z}_{s+t_{z_s}^*}, s+t_{z_s}^*) d\nu_{z^-}(z_{s+t_{z_s}^*}) \end{aligned}$$

so

$$\begin{aligned} U^*(\Phi_z(t), s+t) &= \int_t^{t_{z_s}^*} U^-(\Phi_{\mathbf{z}_s}(u), s+u) \lambda_{z_s}(u) \exp[-\Lambda_{z_s}(u) + \Lambda_{z_s}(t)] du \\ &\quad + \exp[-\Lambda_{z_s}(t_{z_s}^*) + \Lambda_{z_s}(t)] \int_E K_{z^-}(z_{s+t_{z_s}^*}) U^*(\mathbf{z}_{s+t_{z_s}^*}, s+t_{z_s}^*) d\nu_{z^-}(z_{s+t_{z_s}^*}) \\ &= \frac{1}{\exp[-\Lambda_{z_s}(t)]} \int_{[t, t_{z_s}^*]} U^-(\Phi_{\mathbf{z}_s}(u), s+u) \left(\lambda_z(u)\right)^{\mathbb{1}_{t < t_{z_s}^*}} \exp[-\Lambda_z(u)] d\mu_z(t) \end{aligned}$$

This last equality allows to transform (4.8) into (4.9). \square

A.3 Equality (4.12)

Let $s \in [0, t_f]$ and $\mathbf{z}_s \in \mathbf{E}_s$. Remember that equality (4.12) states that if the functions $u \rightarrow U^-(\Phi_{\mathbf{z}_s}(u), s+u)$ and $u \rightarrow \lambda_z(u)$ are continuous almost everywhere on $[0, t_z^*]$, then almost everywhere

$$\frac{\partial U^*(\Phi_{z_s}(v), s+v)}{\partial v} = \left(U^*(\Phi_{z_s}(v), s+v) - U^-(\Phi_{z_s}(v), s+v) \right) \lambda_z(v)$$

Proof. We denote by T the time until the next jump after the trajectory has reached $Z_s = z$. For $0 \leq h < t_z^*$, we define $\tau = \min(h, T)$.

$$\begin{aligned} U^*(\mathbf{z}_s) &= \mathbb{E}[|h(\mathbf{Z})| | \mathbf{Z}_s = \mathbf{z}_s] \\ &= \mathbb{E}\left[\mathbb{E}[|h(\mathbf{Z})| | \mathbf{Z}_{s+\tau}] \middle| \mathbf{Z}_s = \mathbf{z}_s\right] \\ &= \mathbb{E}\left[(\mathbb{1}_{\tau=h} + \mathbb{1}_{\tau < h}) \mathbb{E}[|h(\mathbf{Z})| | \mathbf{Z}_{s+\tau}] \middle| \mathbf{Z}_s = \mathbf{z}_s\right] \\ &= \mathbb{E}\left[\mathbb{1}_{T=h} \mathbb{E}[|h(\mathbf{Z})| | \mathbf{Z}_{s+h} = \Phi_{\mathbf{z}_s}(h)] \middle| \mathbf{Z}_s = \mathbf{z}_s\right] + \mathbb{E}\left[\mathbb{1}_{T < h} \mathbb{E}[|h(\mathbf{Z})| | \mathbf{Z}_{s+T}] \middle| \mathbf{Z}_s = \mathbf{z}_s\right] \\ &= U^*(\Phi_{\mathbf{z}_s}(h), s+h) \mathbb{E}[\mathbb{1}_{T=h} | \mathbf{Z}_s = \mathbf{z}_s] + \mathbb{E}\left[\mathbb{1}_{T < h} U^*(\Phi_{\mathbf{z}_s}(T), s+T) \middle| \mathbf{Z}_s = \mathbf{z}_s\right] \\ &= U^*(\Phi_{\mathbf{z}_s}(h), s+h) \exp[-\Lambda_z(h)] + \\ &\quad \int_0^h \int_E K_{\Phi_z(u)}(z_{s+u}) U^*(\mathbf{z}_{s+u}, s+u) d\nu_{\Phi_z(u)}(z_{s+u}) \lambda_z(u) \exp[-\Lambda_z(u)] du \end{aligned}$$

As $\lambda_z(\cdot)$ is continuous almost everywhere we have that almost everywhere:

$$U^*(\mathbf{z}_s, s) = U^*(\Phi_{\mathbf{z}_s}(h), s + h) (1 - \lambda_z(0)h + o(h)) + \int_0^h U^-(\Phi_{\mathbf{z}_s}(u), s + u) \lambda_z(u) \exp[-\Lambda_z(u)] du$$

As $u \rightarrow U^-(\Phi_{\mathbf{z}_s}(u), s + u) \lambda_z(u)$ is continuous almost everywhere, and we can do a Taylor approximation of the integral, which gives:

$$U^*(\mathbf{z}_s, s) - U^*(\Phi_{\mathbf{z}_s}(h), s + h) = -\lambda_z(0) \cdot h \cdot U^*(\Phi_{\mathbf{z}_s}(h), s + h) + h \cdot U^-(\mathbf{z}_s, s) \lambda_z(0) + o(h)$$

So $u \rightarrow U^*(\Phi_{\mathbf{z}_s}(u), s + u)$ is right-continuous almost everywhere. Therefore $U^*(\Phi_{\mathbf{z}_s}(h), s + h) = U^*(\mathbf{z}_s, s) + o(1)$, and we get:

$$\frac{U^*(\mathbf{z}_s, s) - U^*(\Phi_{\mathbf{z}_s}(h), s + h)}{h} = -\lambda_z(0) U^*(\mathbf{z}_s, s) + U^-(\mathbf{z}_s, s) \lambda_z(0) + o(1)$$

Making h tends to zero we get that $u \rightarrow U^*(\Phi_{\mathbf{z}_s}(u), s + u)$ has a right-derivative in zero. Assuming that $\Phi_{\mathbf{z}_s}(-h)$ is the trajectory defined on $[0, s - h]$ obtained by removing to \mathbf{z}_s its states on $[s - h, s]$. Applying the same kind of reasoning in state $\Phi_{\mathbf{z}_s}(-h)$ instead of \mathbf{z}_s , we would find that the left-derivative exists and is equal to the right-derivative. So for almost every trajectory $\mathbf{z}_s \in \mathbf{E}_s$,

$$\left(\frac{\partial U^*(\Phi_{\mathbf{z}_s}(v), s + v)}{\partial v} \right)_{v=0} = U^*(\Phi_{\mathbf{z}_s}(0), s) \lambda_{z_s}(0) - U^-(\Phi_{\mathbf{z}_s}(0), s) \lambda_{z_s}(0)$$

Applying the same reasoning in a state $\Phi_{z_o}(v)$ instead of z and using the additivity of the flow, we get that almost everywhere:

$$\forall \mathbf{z}_s \in \mathbf{E}_s, v > 0, \quad \frac{\partial U^*(\Phi_{\mathbf{z}_s}(v), s + v)}{\partial v} = U^*(\Phi_{\mathbf{z}_s}(v), s + v) \lambda_{z_s}(v) - U^-(\Phi_{\mathbf{z}_s}(v), s + v) \lambda_{z_s}(v)$$

□

Bibliography

- [1] D. Aristoff. Analysis and optimization of weighted ensemble sampling, *ESAIM: Mathematical Modelling and Numerical Analysis*, **52**(4):1219–1238, 2018.
- [2] D. Aristoff and D.M. Zuckermann. Optimizing weighted ensemble sampling for steady states, *ArXiv e-prints*, June 2018, arXiv: 1806.00860 [math.NA].
- [3] J.L. Beck and K.M. Zuev. Rare-event simulation, *Handbook of uncertainty quantification*:1075–1100, 2017.
- [4] Z. Botev and D. P. Kroese. Global likelihood optimization via the cross-entropy method with an application to mixture models, in *Simulation Conference, 2004. Proceedings of the 2004 Winter*, vol. 1, IEEE, 2004.
- [5] C-E. Bréhier, T. Lelièvre, and M. Rousset. Analysis of adaptive multilevel splitting algorithms in an idealized case, *ESAIM: Probability and Statistics*, 19:361–394, 2015.
- [6] O. Cappé, R. Douc, A. Guillin, J-M. Marin, and C. P Robert. Adaptive importance sampling in general mixture classes, *Statistics and Computing*, 18(4):447–459, 2008.
- [7] R. Carmona and S. Crépey. Particle methods for the estimation of credit portfolio loss distributions, *International Journal of Theoretical and Applied Finance*, 13(04):577–602, 2010.
- [8] V. Caron, A. Guyader, M.M. Zuniga, and B. Tuffin. Some recent results in rare event estimation, in *ESAIM: Proceedings*, vol. 44, pp. 239–259, EDP Sciences, 2014.
- [9] M. Čepin. *Assessment of power system reliability: methods and applications*, Springer, 2011.
- [10] F. Cérou, B. Delyon, A. Guyader, and M. Rousset. On the asymptotic normality of adaptive multilevel splitting, *SIAM/ASA Journal on Uncertainty Quantification*, 7(1):1–30, 2019.
- [11] F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis, *Stochastic Analysis and Applications*, **25**(2):417–443, 2007.
- [12] F. Cérou, F. LeGland, P. Del Moral, and P. Lezaud. Limit theorems for the multilevel splitting algorithm in the simulation of rare events, in *Simulation Conference, 2005 Proceedings of the Winter*, 10–pp, IEEE, 2005.
- [13] J. CC. Chan, P. W. Glynn, D. P. Kroese, et al. A comparison of cross-entropy and variance minimization strategies, *Journal of Applied Probability*, 48:183–194, 2011.

-
- [14] H. Chraibi, A. Dutfoy, T. Galtier, and J. Garnier. Application of the interacting particle system method to piecewise deterministic markov processes used in reliability, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(6):063119, 2019, DOI: 10.1063/1.5081446, eprint: <https://doi.org/10.1063/1.5081446>, URL: <https://doi.org/10.1063/1.5081446>.
- [15] H. Chraibi, A. Dutfoy, T. Galtier, and J. Garnier. On the optimal importance process for piecewise deterministic markov process, *arXiv preprint arXiv:1707.08136*, 2017.
- [16] H. Chraibi, A. Dutfoy, T. Galtier, and J. Garnier. Optimal input potential functions in the interacting particle system method, *arXiv preprint arXiv:1811.10450*, 2018.
- [17] H. Chraibi, J-C. Houbedine, and A. Sibler. Pycatshoo: toward a new platform dedicated to dynamic reliability assessments of hybrid systems, *PSAM congress*, 2016.
- [18] M.H.A. Davis. *Markov Models & Optimization*, vol. 49, CRC Press, 1993.
- [19] M.H.A. Davis. Piecewise-deterministic markov processes: a general class of non-diffusion stochastic models, *Journal of the Royal Statistical Society. Series B (Methodological)*:353–388, 1984.
- [20] P.T. De Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein. A tutorial on the cross-entropy method, *Annals of Operations Research*, 134(1):19–67, 2005.
- [21] P. Del Moral. Feynman-kac formulae, in Springer, 2004.
- [22] P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(3):411–436, 2006.
- [23] P. Del Moral and J. Garnier. Genealogical particle analysis of rare events, *The Annals of Applied Probability*, 15(4):2496–2534, 2005.
- [24] L. Devroye. Sample-based non-uniform random variate generation, in *Proceedings of the 18th conference on Winter simulation*, pp. 260–265, ACM, 1986.
- [25] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering, in *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 64–69, IEEE, 2005.
- [26] R. Douc, A. Guillin, JM. Marin, and CP. Robert. Convergence of adaptive mixtures of importance sampling schemes, *Statistics and Computing*, 18:447–459, 2008.
- [27] F. Dufour, H. Zhang, and B. De Saporta. *Numerical methods for simulation and optimization of piecewise deterministic Markov processes: application to reliability*, John Wiley & Sons, 2015.

-
- [28] P. Dupuis and H. Wang. Importance sampling, large deviations, and differential games, *Stochastics: An International Journal of Probability and Stochastic Processes*, 76(6):481–508, 2004.
- [29] P. Heidelberger. Fast simulation of rare events in queueing and reliability models, *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5(1):43–85, 1995.
- [30] J.D. Hol, T.B. Schon, and F. Gustafsson. On resampling algorithms for particle filters, in *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pp. 79–82, IEEE, 2006.
- [31] D. Jacquemart and J. Morio. Tuning of adaptive interacting particle system for rare event probability estimation, *Simulation Modelling Practice and Theory*, **66**:36–49, 2016.
- [32] S. Juneja and P. Shahabuddin. Rare-event simulation techniques: an introduction and recent advances, *Handbooks in operations research and management science*, 13:291–350, 2006.
- [33] D. P. Kroese, S. Porotsky, and R. Y. Rubinstein. The cross-entropy method for continuous multi-extremal optimization, *Methodology and Computing in Applied Probability*, 8(3):383–407, 2006.
- [34] D.P Kroese, R. Y Rubinstein, I. Cohen, and T. Porotsky S.and Taimre. Cross-entropy method, in *Encyclopedia of Operations Research and Management Science*, pp. 326–333. Springer, 2013.
- [35] I. Kuruganti. Importance sampling for markov chains: computing variance and determining optimal measures, in *Simulation Conference, 1996. Proceedings. Winter*, pp. 273–280, IEEE, 1996.
- [36] P-E. Labeau. A Monte-Carlo estimation of the marginal distributions in a problem of probabilistic dynamics, *Reliability Engineering & System Safety*, 52(1):65–75, 1996.
- [37] P-E. Labeau. Probabilistic dynamics: estimation of generalized unreliability through efficient Monte-Carlo simulation, *Annals of Nuclear Energy*, **23**(17):1355–1369, 1996.
- [38] A. Lee and N. Whiteley. Variance estimation in the particle filter, *Biometrika*, 105(3):609–625, 2018.
- [39] EE. Lewis and F. Böhm. Monte-Carlo simulation of markov unreliability models, *Nuclear engineering and design*, 77(1):49–62, 1984.
- [40] L. Margolin. On the convergence of the cross-entropy method, *Annals of Operations Research*, 134(1):201–214, 2005.

-
- [41] M. Marseguerra and E. Zio. Monte-Carlo approach to psa for dynamic process systems, *Reliability Engineering and System safety*, 52:227–241, 1996.
- [42] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Transition path theory for markov jump processes, *Multiscale Modeling & Simulation*, 7(3):1192–1219, 2009.
- [43] J. Morio, D. Jacquemart, M. Balesdent, and J. Marzat. Optimisation of interacting particle systems for rare event estimation, *Computational Statistics & Data Analysis*, 66:117–128, 2013.
- [44] M Ramakrishnan. Unavailability estimation of shutdown system of a fast reactor by Monte-Carlo simulation, *Annals of Nuclear Energy*, 90:264–274, 2016.
- [45] T. Rolski, H. Schmidli, V. Schmidt, and J. L. Teugels. *Stochastic processes for insurance and finance*, vol. 505, John Wiley & Sons, 2009.
- [46] G. Rubino and B. Tuffin. *Rare event simulation using Monte Carlo methods*, John Wiley & Sons, 2009.
- [47] R. Y. Rubinstein and D. P Kroese. *Simulation and the Monte Carlo method*, vol. 10, John Wiley & Sons, 2016.
- [48] R. Y. Rubinstein and D. P Kroese. *The Cross-Entropy Method*, Springer, New York, 2004, DOI: <https://doi.org/10.1007/978-1-4757-4321-0>.
- [49] A.N. Shiryaev. *Probability, second edition*, Springer-Verlag, volume 95 in graduate texts in mathematics., 1996, DOI: 10.1007/978-1-4757-2539-1.
- [50] D. Siegmund. Importance sampling in the monte carlo study of sequential tests, *The Annals of Statistics*:673–684, 1976.
- [51] S. T. Tokdar and R. E. Kass. Importance sampling: a review, *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60, 2010.
- [52] C. Vergé. *Modèle d’îlots de particules et application en fiabilité*, PhD thesis, ECOLE POLYTECHNIQUE, 2015.
- [53] J. Wouters and F. Bouchet. Rare event computation in deterministic chaotic systems using genealogical particle analysis, *Journal of Physics A: Mathematical and Theoretical*, 49(37):374002, 2016.
- [54] H. Zhang, F. Dufour, Y. Dutuit, and K. Gonzalez. Piecewise deterministic markov processes and dynamic reliability, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 222(4):545–551, 2008.