

# Application of the interacting particle system method to piecewise deterministic Markov processes used in reliability

Hassane CHRAIBI<sup>a</sup>, Anne DUTFOY<sup>a</sup>, Thomas GALTIER<sup>a,b,\*</sup>, Josselin GARNIER<sup>c</sup>

<sup>a</sup>EDF R&D - Département PERICLES, 7 Boulevard Gaspard Monge, 91120 Palaiseau, France

<sup>b</sup> Université Paris-Diderot - Laboratoire de Probabilités Statistique et Modélisation, 75205 Paris Cedex 13, France

<sup>c</sup> Ecole Polytechnique - Centre de Mathématiques Appliquées, 91128 Palaiseau Cedex, France

---

## Abstract

Variance reduction methods are often needed for the reliability assessment of complex industrial systems, we focus on one variance reduction method in a given context, that is the interacting particle system method (IPS) used on piecewise deterministic Markov processes (PDMP) for reliability assessment . The PDMPs are a very large class of processes which benefit from high modeling capacities, they can model almost any Markovian phenomenon that does not include diffusion. In reliability assessment, the PDMPs modeling industrial systems generally involve low jump rates and jump kernels favoring one safe arrival, we call such model a “concentrated PDMP”.

Used on such concentrated PDMPs, the IPS is inefficient and does not always provide a variance reduction. Indeed, the efficiency of the IPS method relies on simulating many different trajectories during its propagation steps, but unfortunately concentrated PDMPs are likely to generate the same deterministic trajectories over and over. We propose an adaptation of the IPS method called IPS+M that reduces this phenomenon. The IPS+M consists in modifying the propagation steps of the IPS, by conditioning the propagation to avoid generating the same trajectories multiple times. We prove that, compared to the IPS, the IPS+M method always provides an estimator with a lower variance. We also carry out simulations on two-components systems that confirm these results.

*Keywords:* Rare event – Reliability assessment – PDMP – Dynamical hybrid system – Sequential Monte-Carlo Samplers – Feymann-Kac particle filters

---

---

\*Corresponding author

Email address: [tgaltier@gmail.com](mailto:tgaltier@gmail.com) (Thomas GALTIER )

**Lead paragraph:** In this paper we introduce and analyze an original version of the interacting particle system algorithm that can be applied to piecewise deterministic Markov processes. This version makes it possible to estimate the probability of a rare event defined in terms of a complex dynamical hybrid system. The consistency and the asymptotic normality of the estimator are proved. The expression of its asymptotic variance allows to quantify the variance reduction. This algorithm can be used for the reliability assessment of complex industrial systems.

## 1. Introduction

For both safety and regulation issues, the reliability of industrial systems has to be assessed. The considered systems (dams or nuclear power plants, for instance) are complex dynamic hybrid systems, so only simulation methods can be reasonably considered to assess their reliability. The failure of such a dynamic hybrid system generally corresponds to a physical variable of the system (temperature, pressure, water level) entering a critical region. The simulation of such a system requires to accurately model the trajectory of the physical variables. The evolution of these physical variables are generally determined by simple differential equations derived from the laws of physics. As the physical context giving the differential equations generally depends on the statuses of the multiple components of the systems (on, off, or failed), the differential equations can change whenever there is a change in the components statuses. To encounter for this hybrid interplay between the discrete process of components' statuses and the continuous evolution of the physical variables, we model the evolution of the state of a system by a piecewise deterministic Markov process (PDMP) [5, 6, 7, 3]. PDMPs are meant to represent a large class of Markovian processes that do not include diffusion, as such, they benefit from high modeling capacity: they can model many industrial systems. For instance, EDF has recently developed the PyCATSHOO toolbox [2], which allows the modeling of dynamic hybrid systems, the main option within this toolbox is to evaluate the dependability criteria of the studied systems by Monte Carlo simulations.

As the industrial systems are highly reliable, their failure is a rare event, and the naive Monte-Carlo simulation method (MC) is computationally too intensive in this context. The objective of our work is to set up new algorithms to accelerate the reliability assessment of such industrial systems. To do so we want to use faster methods, such as variance reduction methods. A variance reduction method is a method that yields an unbiased estimator with a smaller variance than the Monte-Carlo estimator. The estimation being more accurate, we need less simulation runs to reach the desired accuracy, thus we save computational time. The variance reduction is generally achieved by altering the simulation process, and then correcting the bias induced by this modification of simulation process by appropriately weighting each simulation. Several acceleration methods for variance reduction can be proposed, such as importance sampling methods or particle filter methods (also called subset simulations). In this article we focus on one example of particle filter method: the

interacting particle system (IPS) method [9].

Unlike in importance sampling methods, in the IPS method we keep simulating according to the original system. Another difference is that we do not simulate directly the trajectories on the entire observation duration, the observation duration is subdivided into small intervals of time and we simulate the trajectories sequentially one interval of time after the other one. These sequential simulations of the trajectories consist alternating between an exploration step and a selection step. During the exploration step we simulate the trajectories on a small time interval, therefore exploring the most probable trajectories on a short horizon of time. Then we apply a selection step on these trajectories replicating the trajectories which seem close to failure and giving up the less promising trajectories. At the next exploration step, only replicated trajectories are continued, before the next selection, and the next exploration and so on... until each selected trajectory reaches the end of the observation time window. This way the effort of simulation is concentrated on selected trajectories, which have higher chance of becoming a failing trajectory before the end of the observation time window. In the end we get more failing trajectories to fuel our estimation, and, if the selection was well done, the IPS method yields an unbiased estimator with a smaller variance than the MC estimator.

When we try to apply the IPS method in order to estimate the failure probability of such reliable complex hybrid systems, the IPS methods turn out to be inefficient. Indeed the estimation provided by the IPS method often has, in this case, a higher variance than the one of the Monte-Carlo estimator. IPS does not perform well in this context because the application case (a reliable complex hybrid system) makes it hard to conduct the exploration steps of the IPS method efficiently. Indeed, an industrial system is often modeled by what we call a “concentrated PDMP”, which is a PDMP with low jump rates and concentrated jump kernels on boundaries. The typical jump rate is low because it is the sum of the failure rates of the components in working condition and of the repair rates of the failed components which are all very low. These failure and repair rates are very low because the components are reliable and their repairs are slow. The typical jump kernel is concentrated because most of the probability mass of a jump is concentrated on one (safe) output. Indeed the jump kernel on boundaries model the automatic control mechanisms within the system. During such a control mechanism there is a small probability that some component(s) fail on demand but the most likely output is that the system jumps on the safe state aimed by the control mechanism, consequently the probability mass of the jump kernel is concentrated on this output. Due to these characteristics of the model there is a high probability that no component failure or repair occurs during the short exploration time, and with a PDMP it means that all the trajectories are likely to follow the same deterministic paths. So when we explore the trajectory space most simulated trajectories end up being the same one, hence limiting our exploration of the trajectory space. To avoid this pitfall, a particular filter was proposed in [20] that enhances the occurrence of random jumps (failure or repairs) or modifies the occurrence time of the last jump. However the proposed method is limited to a different case of PDMP. This class of PDMP does not include concentrated PDMP, as it contains only PDMP without boundaries which allows continuous jumps kernels, i.e. it does

not allow to model automatic control mechanisms in components. Moreover, an a priori bound on the number of jumps in a time interval is required, but in our case we do not have such information.

In order to adapt the IPS to concentrated PDMPs, we propose instead to use an approach based on the memorization method developed in [16]. The idea is to start the exploration by finding the most likely trajectories continuing each batch of replicated trajectories. Then, we condition the rest of the exploration to avoid these trajectories. As a result, the simulated trajectories have much more chances to differ which improves the quality of the exploration and reduce the variance of the estimator. To correct the bias induced by this modification of the simulation process, we have to modify the weight of each trajectory. We call our adaptation of the IPS to PDMP the IPS+M for sequential Monte-Carlo sampler with memorization method.

The rest of the paper is organized as follows: Section 2 is dedicated to the presentation of our model of the system, Section 3 presents the IPS method and introduces the optimal potential functions for this method, Section 5 presents the IPS+M method which adapts the IPS algorithm for PDMPs with low jump rates, Section 6 explains how to force the differentiation of the trajectories using the memorization method, and finally in Section 7 we illustrate the better efficiency of the IPS+M method on a toy example.

## 2. A model of the system based on a PDMP with discrete jump kernel

### 2.1. The model

We denote by  $Z_t$  the state of the system at time  $t$ .  $Z_t$  is the combination of the physical variables of the system, noted  $X_t$ , and of the statuses of all the components within the system, noted  $M_t$ :  $Z_t = (X_t, M_t)$ . We consider that  $X_t \in \mathbb{R}^d$ , and that  $M_t \in \mathbb{M} = \{On, Off, F\}^{N_c}$  where  $F$  corresponds to a failed status, and  $N_c$  is the number of components in the system. The value of  $M_t$  is sometimes referred as the mode of the system. Here we consider only three categories for the status of a component :  $On, Off, F$ , but it is possible to include more categories as long as the set of the possible modes  $\mathbb{M}$  stays countable.

The process  $Z_t$  is piecewise continuous, and each discontinuity is called a jump. Between two jumps there is no change in the components' statuses, and the dynamics of the physical variables can be expressed thanks to an ordinary differential equation derived from the law of physics:

$$\frac{dX_t}{dt} = F_{M_t}(X_t).$$

We note  $\phi_{(x,m)}(t)$  the solution of this equation when  $X_0 = x$  and  $M_0 = m$ . Then for any time  $s > 0$ , if  $T$  is the time separating  $s$  from the next jump time, we have

$$\forall t \in [0, T), \quad Z_{s+t} = (X_{s+t}, M_s) = (\phi_{(X_s, M_s)}(t), M_s).$$

Similarly, a flow function on the states can be defined. If  $z = (x, m)$ , then we define  $\Phi_z(t) = (\phi_{(x,m)}(t), m)$ , and so

$$\forall t \in [0, T), \quad Z_{t+s} = \Phi_{Z_s}(t). \tag{1}$$

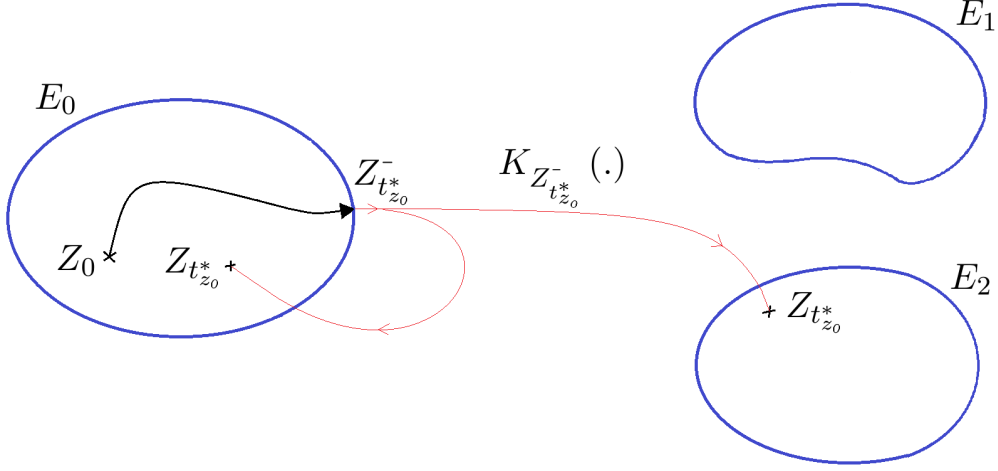


Figure 1: A jump at boundary.

As the physical variables are often continuous, the jumps are essentially used to model changes in the statuses of the components. These jumps can occur for two reasons. Firstly, a jump can correspond to an automatic control mechanism (See Figure 1). In a given mode  $m$ , such mechanism is typically triggered when the physical variables cross some threshold. For each mode  $m$ , we define an open and connected set  $\Omega_m$ , so that these thresholds determine its boundary  $\partial\Omega_m$ . So when  $M_t = m$  the value of the physical variables  $X_t$  is restricted to the set  $\Omega_m \subset \mathbb{R}^d$ . Letting  $E_m = \{(x, m), x \in \Omega_m\}$  be the set of the possible states with mode  $m$ , in terms of state a jump associated to a control mechanism is triggered whenever the state  $Z_{s+t}$  hits the boundary of  $E_m$ . The set of possible states is therefore defined by:

$$E = \bigcup_{m \in \mathbb{M}} E_m.$$

Secondly, jumps can correspond to a spontaneous failure or repair. In such case the jump occurs before  $Z_{s+t}$  hits the boundary of  $E_m$ . The occurrence time of a spontaneous jump is modeled by a jump rate  $\lambda(Z_{s+t})$ . For instance, when we consider that components fail or are repaired one at a time, this jump rate is the sum of the failure rates of the components in working condition and of the repair rates of the broken components. In a more general case, each transition from a mode  $m$  to a mode  $m^+$  has its own rate  $\lambda_{m \rightarrow m^+}(X_{s+t})$ , and we have

$$\lambda(Z_{s+t}) = \sum_{m^+ \in \mathbb{M}} \lambda_{M_s \rightarrow m^+}(X_{s+t}) \quad (2)$$

We define the cumulative jump rate  $\Lambda_z(t)$  by  $\Lambda_z(t) = \int_0^t \lambda(\Phi_z(u)) du$ , so that  $\forall t \in [0, T)$ ,  $\Lambda_{Z_s}(t) = \int_0^t \lambda(Z_{s+u}) du$ . Eventually the cumulative distribution function (cdf) of  $T$  (the time until the next jump starting from a state  $Z_s = z$ ) takes the form:

$$\mathbb{P}(T \leq t | Z_s = z) = \begin{cases} 1 - \exp[-\Lambda_z(t)] & \text{if } t < t_z^*, \\ 1 & \text{if } t \geq t_z^*. \end{cases} \quad (3)$$

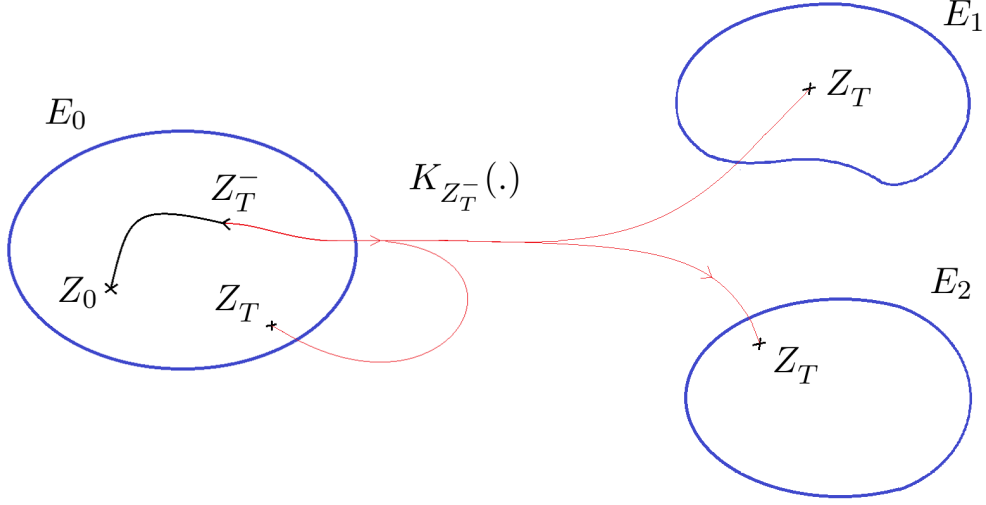


Figure 2: A spontaneous jump.

Here  $t_z^* = \inf\{t > 0, \Phi_z(t) \in \partial E_m\}$  is the time until the flow hits the boundary starting from a state  $z = (x, m)$ . When there is no boundary i.e.  $\{s > 0, \Phi_z(s) \notin E_m\} = \emptyset$ , we take the convention that  $t_z^* = +\infty$ . With this definition, the law of  $T$  has a continuous part associated to spontaneous jumps (spontaneous failures and repairs) and a discrete part associated to forced jumps (control mechanisms). A possible reference measure for  $T$  knowing  $Z_s = z$  is then

$$\forall B \in \mathcal{B}(\mathbb{R}^+), \quad \mu_z(B) = \text{leb}(B \cap (0, t_z^*)) + \mathbb{1}_{t_z^* < \infty} \delta_{t_z^*}(B), \quad (4)$$

where  $\text{leb}(\cdot)$  corresponds to the Lebesgue measure.

If a jump occurs at time  $S$ , then the distribution of the destination of the jump is expressed by a transition kernel  $\mathcal{K}_{Z_S^-}$  where  $Z_S^-$  is the departure state of the jump. If  $\bar{E}$  is the closure of  $E$ , we have  $Z_S^- \in \bar{E}$ . Let  $Z_S^+ \in E$  be the arrival state, and  $\mathcal{B}(E)$  be the Borelian  $\sigma$ -algebra on  $E$ . Then, the law of a jump from a departure state  $z^-$  is defined by:

$$\forall B \in \mathcal{B}(E), \quad \mathbb{P}(Z_T^+ \in B | Z_T^- = z^-) = \mathcal{K}_{z^-}(B). \quad (5)$$

For any departure state  $z^- \in \bar{E}$ , we define  $x_a(z^-, m^+)$  as the arrival for the vector of physical variables when the transition  $m^- \rightarrow m^+$  is triggered from a state  $z^-$ . We define by  $z_a(z^-, m^+) = (x_a(z^-, m^+), m^+)$  the arrival state when the transition  $m^- \rightarrow m^+$  is triggered from a state  $z^-$ . Using this definition, for any departure state  $z^- \in \bar{E}$ , we define the measure  $\nu_{z^-}$  on  $(E, \mathcal{B}(E))$  by:

$$\forall B \in \mathcal{B}(E), \quad \nu_{z^-}(B) = \sum_{m^+ \in \mathbb{M}} \delta_{z_a(z^-, m^+)}(B). \quad (6)$$

The kernel  $\mathcal{K}_{z^-}$  is absolutely continuous with respect to the measure  $\nu_{z^-}$ . Denoting by  $K_{z^-}$  its density with respect to  $\nu_{z^-}$  we have:

$$\mathbb{P}(Z_T^+ \in B | Z_T^- = z^-) = \int_B K_{z^-}(z^+) d\nu_{z^-}(z^+). \quad (7)$$

Note that we only consider systems for which  $\nu_z$  is discrete, whatever the departure state  $z^- \in E$  may be. This hypothesis of discreteness of  $\nu_z$  is mandatory to apply the method presented in Section 5.

If  $\nu_z$  has a Dirac contribution at point  $z$ , then the kernel must satisfy  $K_z(z) = 0$ , so it is not possible to jump on the departure state. In some applications the physical variables are continuous, so that  $K_{z^-}(z)$  is zero whenever  $x^- \neq x$ . In some cases, one might want the model to include renewable and aging components. Then the vector  $X_t$  should include the time since the last renewal of such component, and so the vector  $X_t$  can be discontinuous at the time of a renewal. In both situations, when  $z^-$  is not on the boundary of  $E_{m^-}$ , the jump kernel has this form:

$$\forall z^- \in E, \quad B \in \mathcal{B}(E), \quad \mathcal{K}_{z^-}(B) = \sum_{m^+ \in \mathbb{M}} \frac{\lambda_{m^- \rightarrow m^+}(x^-)}{\lambda(z^-)} \delta_{z_a(z^-, m^+)}(B), \quad (8)$$

$$\text{and} \quad K_{z^-}(z^+) = \frac{\lambda_{m^- \rightarrow m^+}(x^-)}{\lambda(z^-)} \mathbb{1}_{x^+ = x_a(z^-, m^+)}. \quad (9)$$

When  $z^- \in \partial E_{m^-}$ , a control mechanism is triggered : some components are required to turn on, or to turn off, so that the system reaches a desired state  $z_c$ . The transition to this state is usually very likely, so when  $z^- \in \partial E_{m^-}$ , the jump kernel  $K_{z^-}$  tends to concentrate all or a big portion of the probability mass on this state  $z_c$ . In industrial systems there can be components which have a small probability to fail when they are asked to turn on. This phenomenon is referred as a failure on demand. We denote by  $\kappa_i(z^-)$  the probability that the  $i^{\text{th}}$  component fails on demand when the control is triggered. We denote by  $fod(i, m^-, m)$  the indicator being equal to one if the  $i^{\text{th}}$  component fails on demand during the transition from  $m^-$  to  $m$ , and to zero otherwise. We also define  $ask(z^-, z^+)$  as the set gathering the indices of the components supposed to turn on during the control mechanism triggered from  $z^-$  to  $z^+$ . Finally, when  $z^- \in \partial E_{m^-}$ , we have that:

$$K_{z^-}(z^+) = \mathbb{1}_{x^+ = x_a(z^-, m^+)} \prod_{i \in ask(z^-, z^+)} (\kappa_i(z^-))^{fod(i, m^-, m^+)} (1 - \kappa_i(z^-))^{(1 - fod(i, m^-, m^+))}. \quad (10)$$

To generate  $\mathbf{Z}_t = (Z_s)_{s \in [0, t]}$  a trajectory of the states of the system, one can repeat the following steps: Starting with  $s = 0$ ,

1. Given a starting state  $Z_s = z$ , generate  $T$  the time until the next jump using (3),
2. Follow the flow  $\Phi$  until  $s + T$  using (1), and set the departure state of the next jump as being  $Z_T^- = \Phi_z(T)$ ,
3. generate  $Z_{s+T}$  the arrival state of the next jump using  $K_{Z_T^-}$
4. repeat starting with  $s = s + T$  until one gets a trajectory of size  $t$ .

Defined in this way, the process  $Z_t$  is Markovian [4].

## 2.2. Law of trajectory

Let us denote by  $n(\mathbf{Z}_t)$  the number of jumps in the trajectory  $\mathbf{Z}_t$ , and by  $S_k$  the time of the  $k^{\text{th}}$  jump in the trajectory  $\mathbf{Z}_t$  (with the convention that  $S_0 = 0$  and  $S_{n(\mathbf{Z}_t)+1} = t$ ).  $T_k = S_{k+1} - S_k$  denotes the time between two consecutive jumps. We define the  $\sigma$ -algebra  $\mathcal{S}_t$  as the  $\sigma$ -algebra generated by the sets in  $\bigcup_{n \in \mathbb{N}^*} \mathcal{B}\left(\left\{(z_k, t_k)_{k \leq n} \in (E \times \mathbb{R}_+^*)^n, \sum_{i=0}^n t_i = t\right\}\right)$ , where  $\mathcal{B}(\cdot)$  indicates the Borelians of a set. Letting  $\Theta_t : \mathbf{Z}_t \rightarrow ((Z_k, T_k))_{0 \leq k \leq n(\mathbf{Z}_t)}$  be the application giving the skeleton of a trajectory, we can define the law of trajectories as an image law through  $\Theta_t$ . We have that for  $B \in \mathcal{S}_t$ :

$$\begin{aligned} \mathbb{P}_{z_0}(\mathbf{Z}_t \in \Theta_t^{-1}(B)) &= \int_B \prod_{k=0}^n (\lambda_{z_k}(t_k))^{\mathbb{1}_{t_k < t_n^*}} \exp\left[-\Lambda_{z_k}(t_k)\right] \prod_{k=1}^n K_{z_k^-}(z_k) \\ &\quad \times d\delta_{t_n^*}(t_n) d\nu_{z_n^-}(z_n) d\mu_{z_{n-1}}(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{z_0}(t_0), \end{aligned} \quad (11)$$

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

In the rest of the paper we will denote  $\mathbf{E}_t$  the set of trajectories of size  $t$  that satisfy (1), and by  $\mathcal{M}(\mathbf{E}_t)$  the set of bounded  $(\Theta_t^{-1}(\mathcal{S}_t), \mathbf{E}_t)$ -measurable functions.

## 2.3. A hybrid reference measure for the trajectories of PDMP

As they involve hybrid reference measures for the times between jumps, PDMPs are very degenerate processes. As a result, some of their realizations have a strictly positive probability to be simulated, but some other do not. The trajectories that concentrate a part of the probability mass on their own, i.e. that verify  $\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) > 0$ , are called the preponderant trajectories. For example, assuming that  $\lambda$  is positive, if the trajectory  $\mathbf{z}_t$  involves no jump, it is preponderant because we have  $\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) = 1 - \exp(-\Lambda_{z_0}(t)) > 0$ . Also, as we consider that the reference measures  $\nu_{z^-}$  are discrete, there can be an other type of preponderant trajectories, which are the trajectories that only jump on the boundaries  $\partial E_m$ . Indeed, such trajectories  $\mathbf{z}_t$  would verify

$$\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) = \prod_{k=0}^n \exp\left[-\Lambda_{z_{s_k}}(t_{z_{s_k}}^*)\right] \prod_{k=1}^n K_{z_{s_k}^-}(z_{s_k}) > 0, \quad (12)$$

where  $s_k$  is the time of the  $k^{\text{th}}$  jump in  $\mathbf{z}_t$ . Conversely, some realizations can be considered as negligible as they verify  $\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) = 0$ . These trajectories are the ones that involve a spontaneous jump, i.e. a jump starting from the interior of a set  $E_m$ .

This can be better understood by looking at the equation (11) which shows that the measure  $\zeta_{z_0, t}$  defined by :

$$\forall B \in \mathcal{S}_t, \quad \zeta_{z_0, t}(\Theta^{-1}(B)) = \int_{(z_k, t_k)_{k \leq n} \in B} d\delta_{t_n^*}(t_n) d\nu_{z_n^-}(z_n) d\mu_{z_{n-1}}(t_{n-1}) \dots d\nu_{z_1^-}(z_1) d\mu_{z_0}(t_0) \quad (13)$$

where  $z_j^- = \Phi_{z_{j-1}}(t_{j-1})$ , and  $t_n^* = t - \sum_{i=0}^{n-1} t_i$  is a reference measure for the law of trajectories. This reference measure highlights the fact that the trajectories with no jump or with



only jumps on boundaries can concentrate some probability mass on their own, because they refer to a Dirac contribution of the measure  $\zeta_{z_0,t}$ . Indeed for such trajectories the times between two consecutive jumps verify  $t_k = t_{z_{s_k}}^* \quad \forall k < n$ , and so, they always refer to the discrete part of the measures  $\mu_{z_{k-1}} \quad \forall k < n$ , therefore such trajectories are related to the discrete part of  $\zeta_{z_0,t}$ . Equation (13) also shows that the remaining probability mass is distributed continuously among the trajectories with at least one spontaneous jump. Indeed, in a negligible trajectory, if for example the  $k + 1^{\text{th}}$  jump is a spontaneous jump, then  $t_k$  relates to the continuous part of the reference measure  $\mu_{z_{k-1}}$ .

#### 2.4. Concentrated PDMP

For reliability assessment of a highly reliable system one often models the system by a PDMP with low jump rates and concentrated jump kernels on the boundaries. Indeed, the components of the system are often reliable and their repair takes time, hence the low jump rates, and 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). We call this kind of PDMP a concentrated PDMP, mainly because the law of one trajectory concentrates a big part of its probability mass.

This trajectory happens to be the trajectory with no failure and no 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

$$\mathbb{P}_{z_{s_k}} \left( T_k = t_{z_{s_k}}^* \right) = \exp \left[ - \Lambda_{z_{s_k}} \left( t_{z_{s_k}}^* \right) \right] \simeq 1.$$

So only jumps on boundaries are likely, and when the process hits a boundary  $\partial E_m$ , the arrival state aimed by the control mechanism is very likely. Denoting by  $z_{s_k}$  this state for a  $k$ -th jump we have :

$$K_{z_{s_k}}^-(z_{s_k}) \simeq 1.$$

So if  $\mathbf{z}_t$  is a trajectory with no failure and no repair of reasonable size we have:

$$\mathbb{P}(\mathbf{Z}_t = \mathbf{z}_t) = \prod_{k=0}^n \exp \left[ - \Lambda_{z_{s_k}} \left( t_{z_{s_k}}^* \right) \right] \prod_{k=1}^n K_{z_{s_k}}^-(z_{s_k}) \simeq 1. \quad (14)$$

In this article we adapt the IPS method for concentrated PDMPs. It is interesting to make the connection between our work and the modified particle filter developed in [12] which can be applied to the related context of discrete time processes with discrete state spaces.

#### 2.5. Reliability assessment

Let  $t_f$  be an observation time. We denote by  $\mathbf{E}_{t_f}$  the set of trajectories of size  $t_f$  that verify (1), and  $h \in \mathcal{M}(\mathbf{E}_{t_f})$ . The methods presented in this article can be used for the estimation of any quantity  $p_h$  defined by:

$$p_h = \mathbb{E}[h(\mathbf{Z}_{t_f})].$$

We are interested in estimating the probability, noted  $p_{\mathcal{D}}$ , that the system fails before the final observation time  $t_f$  knowing it was initiated in a state  $z_0$ . Letting  $\mathcal{D}$  be the set of trajectories of length  $t_f$  which pass through the critical region  $D \subset E$ , we have

$$p_{\mathcal{D}} = \mathbb{P}_{z_0}(\mathbf{Z}_{t_f} \in \mathcal{D}) = \mathbb{E}_{z_0}[\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{t_f})].$$

Although our application relates to the case where  $h = \mathbb{1}_{\mathcal{D}}$ , the IPS method will be presented with an arbitrary (bounded) function  $h$ .

### 3. The IPS method

#### 3.1. A Feynman-Kac model

For a measure  $\eta$  and a bounded measurable function  $f$  we note  $\eta(f) = \int f d\eta$ . For a measure  $\eta$  and a kernel  $V$ ,  $\eta V$  denotes the measure such that  $\eta V(h) = \int \int h(y)V(dy|x)d\eta(x)$ , and for a bounded measurable function  $f$ ,  $V(f)$  is the function such that  $V(f)(x) = \int h(y)V(dy|x)$ .

Consider a subdivision of the interval  $[0, t_f]$  into  $n$  sub-intervals of equal lengths, noted  $[\tau_k, \tau_{k+1})$ , and such that  $0 = \tau_0 < \tau_1 < \dots < \tau_{n-1} < \tau_n = t_f$ . Let  $V_k$  be the Markovian transition measure extending a trajectory of size  $\tau_k$  into a trajectory of size  $\tau_{k+1}$ , such that

$$\forall \mathbf{z}_{\tau_k} \in \mathbf{E}_{\tau_k}, B \in \mathcal{E}_{\tau_{k+1}}, \quad \mathbb{P}(\mathbf{Z}_{\tau_{k+1}} \in B | \mathbf{Z}_{\tau_k} = \mathbf{z}_{\tau_k}) = V_k(B | \mathbf{z}_{\tau_k}). \quad (15)$$

For each  $k < n$  we denote  $G_k$  the potential function on  $\mathbf{E}_{\tau_k}$ , such that:

$$\forall \mathbf{z}_k \in \mathbf{E}_k, \quad G_k(\mathbf{z}_k) \geq 0. \quad (16)$$

These potential functions are the main inputs of the method. The choice of the potential functions is important, because it will ultimately determine the variance of the estimator of  $p_h$  provided by the IPS method. A good choice depends on the system and on the target function  $h$ . The potential functions are used to define the target probability measures  $\tilde{\eta}_k$  for each  $k \leq n$ , such that:

$$\tilde{\eta}_k(d\mathbf{z}_{\tau_k}) \propto \prod_{s=0}^k G_s(\mathbf{z}_{\tau_s}) \prod_{s=0}^{k-1} V_s(d\mathbf{z}_{\tau_{s+1}} | \mathbf{z}_{\tau_s}), \quad (17)$$

or equivalently

$$\forall B \in \mathcal{E}_{\tau_k}, \quad \tilde{\eta}_k(B) = \frac{\mathbb{E} \left[ \mathbb{1}_B(\mathbf{Z}_{\tau_k}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}{\mathbb{E} \left[ \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}. \quad (18)$$

Originally the IPS method comes from filtering methods. Filtering methods aim at estimating the target measures and, in these methods, the potential functions are chosen so that the  $\eta_k$  match the target measures. But in our cases we have no interest in estimating the target measures, we only want to estimate  $p_h$ . So the potential functions can be chosen

more freely. They are used to propose a probabilistic representation of  $p_h$  in terms of a selection+mutation dynamics, which makes it possible to build an estimator of  $p_h$  with a reduced variance.

We define the propagated target measures  $\eta_k$  such that  $\eta_0 = \tilde{\eta}_0$  and for  $k \geq 0$ ,  $\eta_{k+1} = \tilde{\eta}_k V_k$ . We have :

$$\eta_{k+1}(d\mathbf{z}_{\tau_{k+1}}) \propto \prod_{s=0}^k G_s(\mathbf{z}_{\tau_s}) \prod_{s=0}^k V_s(d\mathbf{z}_{\tau_{s+1}} | \mathbf{Z}_{\tau_s}), \quad (19)$$

or equivalently

$$\forall B \in \mathcal{E}_{\tau_{k+1}}, \quad \eta_{k+1}(B) = \frac{\mathbb{E} \left[ \mathbb{1}_B(\mathbf{Z}_{\tau_{k+1}}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}{\mathbb{E} \left[ \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right]}. \quad (20)$$

For  $k = 0$  we consider that  $\eta_0 = \delta_0$ , but the methods would still be valid if we had  $\eta_0 \neq \delta_0$ . We define  $Q_k$  such that for  $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$ ,

$$Q_k(f)(\mathbf{Z}_{\tau_k}) = \int_{\mathbf{E}_{\tau_{k+1}}} f(\mathbf{z}_{\tau_{k+1}}) \mathbf{V}_k(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}) G_k(\mathbf{Z}_{\tau_k})$$

and set  $Q_{k,n} = Q_k Q_{k+1} \dots Q_n$ . Let  $\Psi_k$  be the application that transforms a measure  $\eta$  defined on  $\mathbf{E}_{\tau_k}$  into a measure  $\Psi_k(\eta)$  defined on  $\mathbf{E}_{\tau_k}$  as follows:

$$\Psi_k(\eta)(f) = \frac{\int G_k(\mathbf{z}) f(\mathbf{z}) d\eta(\mathbf{z})}{\eta(G_k)}. \quad (21)$$

We say that  $\Psi_k(\eta)$  gives the selection of  $\eta$  through the potential  $G_k$ . Notice that  $\tilde{\eta}_k$  is the selection of  $\eta_k$  as  $\tilde{\eta}_k = \Psi_k(\eta_k)$ . The target distributions can therefore be built according to the following pattern, in which a propagation step follows a selection step:

$$\eta_k \xrightarrow{\Psi_k} \tilde{\eta}_k \xrightarrow{V_k} \eta_{k+1}.$$

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

$$\tilde{\gamma}_k(f) = \mathbb{E} \left[ f(\mathbf{Z}_{\tau_k}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right] \quad \text{and} \quad \tilde{\eta}_k(f) = \frac{\tilde{\gamma}_k(f)}{\tilde{\gamma}_k(1)}, \quad (22)$$

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

$$\gamma_{k+1}(f) = \mathbb{E} \left[ f(\mathbf{Z}_{\tau_{k+1}}) \prod_{s=0}^k G_s(\mathbf{Z}_{\tau_s}) \right] \quad \text{and} \quad \eta_{k+1}(f) = \frac{\gamma_{k+1}(f)}{\gamma_{k+1}(1)}. \quad (23)$$

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:

$$p_h = \gamma_n(f_h) = \eta_n(f_h) \prod_{k=0}^{n-1} \eta_k(G_k). \quad (24)$$

### 3.2. The IPS algorithm and its estimators

The IPS method provides an algorithm to generate weighted samples which approximate the probability measures  $\eta_k$  and  $\tilde{\eta}_k$  respectively for each step  $k$ . For the sample approximating  $\eta_k$ , we denote  $\mathbf{Z}_{\tau_k}^j$  the  $j^{\text{th}}$  trajectory and  $W_k^j$  its weight. Respectively, for the sample approximating  $\tilde{\eta}_k$ , we denote  $\tilde{\mathbf{Z}}_{\tau_k}^j$  the  $j^{\text{th}}$  trajectory and  $\tilde{W}_k^j$  its associated weight. For simplicity reasons, in this paper, we consider that the samples all contain  $N$  trajectories, but it is possible to modify the sample size at each step, as illustrated in [17]. The empirical approximations of  $\eta_k$  and  $\tilde{\eta}_k$  are denoted by  $\eta_k^N$  and  $\tilde{\eta}_k^N$  and are defined by:

$$\tilde{\eta}_k^N = \sum_{i=1}^N \tilde{W}_k^i \delta_{\tilde{\mathbf{Z}}_{\tau_k}^i} \quad \text{and} \quad \eta_k^N = \sum_{i=1}^N W_k^i \delta_{\mathbf{Z}_{\tau_k}^i}. \quad (25)$$

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

$$\tilde{\eta}_k^N(f) = \sum_{i=1}^N \tilde{W}_k^i f(\tilde{\mathbf{Z}}_{\tau_k}^i) \quad \text{and} \quad \eta_k^N(f) = \sum_{i=1}^N W_k^i f(\mathbf{Z}_{\tau_k}^i). \quad (26)$$

By plugging these estimations into equations (22) and (22), we get estimations for the unnormalized distributions. Denoting by  $\tilde{\gamma}_k^N$  and  $\gamma_k^N$  these estimations, for all  $k \leq n$  and  $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$ , we have:

$$\tilde{\gamma}_k^N(f) = \tilde{\eta}_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s) \quad \text{and} \quad \gamma_k^N(f) = \eta_k^N(f) \prod_{s=0}^{k-1} \eta_s^N(G_s). \quad (27)$$

Plugging the estimations  $\eta_k^N$  into equation (24), we get an estimator  $\hat{p}_h$  of  $p_h$  defined by:

$$\hat{p}_h = \eta_n^N(f_h) \prod_{k=0}^{n-1} \eta_k^N(G_k). \quad (28)$$

The algorithm builds the samples sequentially, alternating between a selection step and a propagation step. The  $k^{\text{th}}$  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: 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 denote by  $A_k^j$  the ancestor index of the  $j^{\text{th}}$  trajectory in the selected sample, such that  $\tilde{\mathbf{Z}}_k^j = \mathbf{Z}_{\tau_k}^{A_k^j}$ . We let  $\tilde{N}_k^j = \text{card}\{i, A_k^i = j\}$  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. So the selected sample focuses on trajectories that will have a greater impact on the estimations of the next distributions once extended.

<p><b>Initialization :</b> <math>k = 0</math>, <math>\forall j = 1..N</math>, <math>\mathbf{Z}_0^j \stackrel{i.i.d.}{\sim} \eta_0</math> and <math>W_0^j = \frac{1}{N}</math>, and <math>\tilde{W}_0^j = \frac{G_0(\mathbf{Z}_0^j)}{\sum_s G_0(\mathbf{Z}_0^s)}</math></p> <p><b>while</b> <math>k &lt; n</math> <b>do</b></p> <p style="padding-left: 20px;"><b>Selection:</b></p> <p style="padding-left: 20px;">Sample <math>(\tilde{N}_k^j)_{j=1..N} \sim \text{Mult}(N, (\tilde{W}_k^j)_{j=1..N})</math></p> <p style="padding-left: 20px;"><math>\forall j := 1..N</math>, <math>\tilde{W}_k^j := \frac{1}{N}</math></p> <p style="padding-left: 20px;"><b>Propagation :</b></p> <p style="padding-left: 20px;"><b>for</b> <math>j := 1..N</math> <b>do</b></p> <p style="padding-left: 40px;">Sample <math>\mathbf{Z}_{\tau_{k+1}}^j</math> from <math>\mathbf{V}_{k+1}(\cdot   \tilde{\mathbf{Z}}_{\tau_k}^j)</math></p> <p style="padding-left: 40px;">set <math>W_{k+1}^j = \tilde{W}_k^j</math></p> <p style="padding-left: 20px;"><b>for</b> <math>i := 1..N</math> <b>do</b></p> <p style="padding-left: 40px;">Set <math>\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)}</math></p> <p style="padding-left: 20px;"><b>if</b> <math>\forall j</math>, <math>\tilde{W}_{k+1}^j = 0</math> <b>then</b></p> <p style="padding-left: 40px;"><math>\forall q &gt; k</math>, set <math>\eta_q^N = \tilde{\eta}_q^N = 0</math> and Stop</p> <p style="padding-left: 20px;"><b>else</b></p> <p style="padding-left: 40px;"><math>k := k + 1</math></p>
--

Table 1: IPS algorithm

If one specifies potential functions that are not positive, there can be a possibility that at a step  $k$  we get  $\forall j, G_k(\mathbf{Z}_{\tau_k}^j) = 0$ , and so the probability for resampling cannot be defined. When this is the case, the algorithm stops and we consider that  $\forall s \geq k$  the measures  $\tilde{\eta}_s^N$  and  $\eta_s^N$  are equal to the null measure.

Then the  $k^{\text{th}}$  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  $\tilde{\mathbf{Z}}_k^j$  on the interval  $[\tau_k, \tau_{k+1})$  using the transition kernel  $V_k$ . 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 1.

For the sake of simplicity, we will make the following assumption:  $\exists \varepsilon_1, \varepsilon_2 \in \mathbb{R}^+$  such that  $\forall \mathbf{Z}_{\tau_k} \in \mathbf{E}_{\tau_k}$ :

$$\varepsilon_1 > G_k(\mathbf{Z}_{\tau_k}) > \varepsilon_2 > 0, \quad ((G))$$

**Theorem 1.** *When (G) is verified the estimator (28) is unbiased and strongly consistent.*

The proof of theorem 1 follows from Theorems 7.4.2 and 7.4.3 in [8].

**Theorem 2.** *When (G) is verified:*

$$\sqrt{N}(\hat{p}_h - p_h) \xrightarrow[N \rightarrow \infty]{d} \mathcal{N}(0, \sigma_{IPS,G}^2), \quad (29)$$

where

$$\sigma_{IPS,G}^2 = \sum_{k=0}^{n-1} \gamma_k(1)^2 \eta_k \left( [Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h)]^2 \right) \quad (30)$$

$$= \sum_{k=0}^{n-1} \left\{ \mathbb{E}_{z_0} \left[ \prod_{i=0}^{k-1} G_i(\mathbf{Z}_{\tau_i}) \right] \mathbb{E}_{z_0} \left[ \mathbb{E}[h(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_k}]^2 \prod_{s=0}^{k-1} G_s^{-1}(\mathbf{Z}_{\tau_s}) \right] - p_h^2 \right\}. \quad (31)$$

A proof of this CLT can be found in [8] chapter 9 at the theorem 9.3.1 . For the estimation of the variance  $\sigma_{IPS,G}^2$  we refer the reader to [17].

### 3.3. Classical improvements of the IPS method: the SMC method, and its alternative resampling

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 into the estimation (see (31)). So we would like to trigger them only when it is judicious. Typically, not 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 Sequential Monte Carlo (SMC) algorithm with adaptive resampling presented in Table 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:

$$ESS_k = \frac{\left( \sum_{j=0}^N W_k^j G_k(\mathbf{Z}_{\tau_k}^j) \right)^2}{\sum_{i=0}^N \left( W_k^i G_k(\mathbf{Z}_{\tau_k}^i) \right)^2}. \quad (32)$$

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  $ESS_k \leq eN$  where  $e \in [0, 1]$  is a tuning parameter.

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 [11, 14]. There are also recent theoretical results on the performance of different resampling schemes [13]. 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 [19] for instance.

<p><b>Initialization :</b> <math>k = 0</math>, <math>\forall i := 1..N</math>, <math>\mathbf{Z}_0^i = (z_0)</math> and <math>W_0^i = \frac{1}{N}</math>, and <math>\tilde{W}_0^i = \frac{G_0(\mathbf{Z}_0^i)}{\sum_j G_0(\mathbf{Z}_0^j)}</math></p> <p><b>while</b> <math>k &lt; n</math> <b>do</b></p> <p>    <b>Selection:</b></p> <p>    <b>if</b> <math>ESS_k \leq eN</math> <b>then</b></p> <p>          Sample <math>(\tilde{N}_k^j)_{j=1..N} \sim Mult(N, (\tilde{W}_k^j)_{j=1..N})</math> and set <math>\forall i = 1..n</math>, <math>\tilde{W}_k^i := \frac{1}{N}</math></p> <p>    <b>else</b></p> <p>          <b>for</b> <math>i := 1..N</math> <b>do</b></p> <p>              set <math>\tilde{\mathbf{Z}}_{\tau_k}^i := \mathbf{Z}_{\tau_k}^i</math></p> <p>    <b>Propagation :</b></p> <p>    <b>for</b> <math>i := 1..N</math> <b>do</b></p> <p>          Sample <math>\mathbf{Z}_{\tau_{k+1}}^j</math> from <math>\mathbf{V}_{k+1}(\cdot   \tilde{\mathbf{Z}}_{\tau_k}^j)</math></p> <p>          set <math>W_{k+1}^i = \tilde{W}_k^i</math></p> <p>    <b>for</b> <math>i := 1..N</math> <b>do</b></p> <p>          Set <math>\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)}</math></p> <p>    <b>if</b> <math>\forall j</math>, <math>\tilde{W}_{k+1}^j = 0</math> <b>then</b></p> <p>          <math>\forall q &gt; k</math>, set <math>\eta_q^N = \tilde{\eta}_q^N = 0</math> and Stop</p> <p>    <b>else</b></p> <p>          <math>k := k + 1</math></p>
--

Table 2: SMC algorithm with adaptive resampling steps

#### 4. Choice of the potential functions

Note that the variance of  $\hat{p}_h$  depend on the number of subdivisions and on the choice of the potential functions. We display here an important hint on how to select potential functions that yield a small variance. Indeed, the theoretical expressions of the potential functions that minimize the asymptotic variance of the IPS estimator are known [1].

**Theorem 3.** For  $k \geq 1$ , let  $G_k^*$  be defined by:

$$G_k^*(\mathbf{z}_{\tau_k}) = \sqrt{\frac{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_{k+1}}\right]^2|\mathbf{Z}_{\tau_k} = \mathbf{z}_{\tau_k}\right]}{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_k}\right]^2|\mathbf{Z}_{\tau_{k-1}} = \mathbf{z}_{\tau_{k-1}}\right]}} \quad (33)$$

if  $\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_k}\right]^2|\mathbf{Z}_{\tau_{k-1}} = \mathbf{z}_{\tau_{k-1}}\right] \neq 0$ , and  $G_k^*(\mathbf{z}_{\tau_k}) = 0$  otherwise. For  $k = 0$ , we define

$$G_0^*(\mathbf{z}_{\tau_0}) = \sqrt{\mathbb{E}\left[\mathbb{E}\left[h(\mathbf{Z}_{\tau_n})|\mathbf{Z}_{\tau_1}\right]^2|\mathbf{Z}_{\tau_0} = \mathbf{z}_{\tau_0}\right]}. \quad (34)$$

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

$$\begin{aligned} \sigma_{IPS, G^*}^2 &= \mathbb{E} \left[ \mathbb{E} [h(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_0}]^2 \right] - p_h^2 \\ &+ \sum_{k=1}^n \left\{ \mathbb{E} \left[ \sqrt{\mathbb{E} \left[ \mathbb{E} [h(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_k}]^2 | \mathbf{Z}_{\tau_{k-1}} \right]} \right]^2 - p_h^2 \right\}. \end{aligned} \quad (35)$$

For the reliability assessment case, the optimal target distributions have the form:

$$\tilde{\eta}_k^*(d\mathbf{z}_{\tau_k}) \propto \sqrt{\mathbb{E} \left[ \mathbb{E} [\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_{k+1}}]^2 | \mathbf{Z}_{\tau_k} = \mathbf{z}_{\tau_k} \right]} \prod_{s=1}^k V_s(d\mathbf{z}_{\tau_s} | \mathbf{z}_{\tau_{s-1}}), \quad (36)$$

and the potential functions would be defined by (33) and (34), taking  $h = \mathbb{1}_{\mathcal{D}}$ . As we do not have closed-form expressions of the functions  $\mathbf{z}_{\tau_k} \rightarrow \sqrt{\mathbb{E} \left[ \mathbb{E} [\mathbb{1}_{\mathcal{D}}(\mathbf{Z}_{\tau_n}) | \mathbf{Z}_{\tau_{s+1}}]^2 | \mathbf{Z}_{\tau_s} = \mathbf{z}_{\tau_s} \right]}$ , we propose to use instead a parametric approximation of these expectations based on our knowledge of the system and denoted by  $U_\alpha(\mathbf{Z}_{\tau_s})$  so that we take

$$\tilde{\eta}_k(d\mathbf{Z}_{\tau_k}) \propto U_\alpha(\mathbf{Z}_{\tau_k}) \prod_{s=1}^k V_s(d\mathbf{Z}_{\tau_s} | \mathbf{Z}_{\tau_{s-1}}) \quad (37)$$

$$\text{and } \forall s > 0, \quad G_s(\mathbf{Z}_{\tau_s}) = \frac{U_\alpha(\mathbf{Z}_{\tau_s})}{U_\alpha(\mathbf{Z}_{\tau_{s-1}})} \quad \text{and} \quad G_0(\mathbf{Z}_{\tau_0}) = U_\alpha(\mathbf{Z}_{\tau_0}). \quad (38)$$

For a system including similar components in parallel redundancy, we propose to set  $U_\alpha(\mathbf{Z}_{\tau_s}) = 1$  when  $\mathbf{Z}_{\tau_s}$  has already reached the failure region once, and to set  $U_\alpha(\mathbf{Z}_{\tau_s}) = \exp[-\alpha(b(\mathbf{Z}_{\tau_s}) + 1)^2] L(\tau_s)$  otherwise, where  $L$  is a positive function, and  $b(Z)$  indicates the number of working components within a state  $Z$ . Here  $\alpha$  is a parameter tuning the strength of the selection.

## 5. The IPS+M method for concentrated PDMPs

### 5.1. The issue with PDMP modeling a reliable system

When it is used on a reliable system and therefore on a concentrated PDMP (see Section 2.4), 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  $\eta_{k+1}^N$  of  $\eta_{k+1}$ . 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  $\eta_{k+1}$ , which eventually deteriorates the quality of the estimator  $\hat{p}_{\mathcal{D}}$  or  $\hat{p}_h$ .

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 (25) in the following way

$$\tilde{\eta}_k^N = \sum_{i=1}^N \tilde{W}_k^i \delta_{\tilde{\mathbf{Z}}_{\tau_k}^i} = \frac{1}{N} \sum_{j=1}^N \tilde{N}_k^j \delta_{\mathbf{Z}_{\tau_k}^j} \quad (39)$$

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  $V_k(\cdot | \mathbf{Z}_{\tau_k}^j)$ . (For all index  $i$  such that  $A_k^i = j$  the trajectory  $\tilde{\mathbf{Z}}_{\tau_k}^i$  is extended with the kernel  $V_k(\cdot | \mathbf{Z}_{\tau_k}^j)$ ). As the kernel  $V_k(\cdot | \mathbf{Z}_{\tau_k}^j)$  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  $\tau_{k+1}$  without spontaneous jump or failure concentrates the mass of the kernel  $V_k(\cdot | \mathbf{Z}_{\tau_k}^j)$ . Indeed, at this point we have :

$$V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) = \mathbb{P}(\mathbf{Z}_{\tau_{k+1}} = \mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k} = \mathbf{Z}_{\tau_k}^j) \simeq 1. \quad (40)$$

Therefore each of the trajectories  $\tilde{\mathbf{Z}}_{\tau_k}^i = \mathbf{Z}_{\tau_k}^j$  in a cluster tends to be extended in  $\mathbf{a}_{\tau_{k+1}}^{k,j}$ . 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  $h = \mathbb{1}_{\mathcal{D}}$ . So the potential of a trajectory  $G_{k+1}(\mathbf{Z}_{\tau_{k+1}})$  should be high if its final state  $\mathbf{Z}_{\tau_{k+1}}$  is more degraded than the state  $\mathbf{Z}_{\tau_k}$ . This generally implies that  $\mathbf{Z}_{\tau_{k+1}}$  includes at least one component failure between  $\tau_k$  and  $\tau_{k+1}$ . As the preponderant trajectories  $\mathbf{a}_{\tau_{k+1}}^{k,j}$  do not contain failure between  $\tau_k$  and  $\tau_{k+1}$  they generally are associated with low potential values.

The segment of  $\mathbf{a}_{\tau_{k+1}}^{k,j}$  on  $(\tau_k, \tau_{k+1}]$  relates to a Dirac contribution of the measure  $\zeta_{\mathbf{Z}_{\tau_k}^j, \tau_{k+1} - \tau_k}$ . We can, therefore, decompose the expected propagation of the trajectory  $\mathbf{Z}_{\tau_k}^j$  in this way:

$$\delta_{\mathbf{Z}_{\tau_k}^j} V_k(f) = f(\mathbf{a}_{\tau_{k+1}}^{k,j}) V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) + \int_{\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}} f(\mathbf{z}_{\tau_{k+1}}) V_k(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j), \quad (41)$$

where  $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$ . And the expected propagation of  $\eta_k^N$  would be:

$$\begin{aligned}
\tilde{\eta}_k^{\tilde{N}} V_k(f) &= \sum_{j=1}^{\tilde{N}_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} \delta_{\mathbf{z}_k^j} V_k(f) \\
&= \sum_{j=1}^{\tilde{N}_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} f(\mathbf{a}_{\tau_{k+1}}^{k,j}) V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) + \frac{\tilde{N}_k^j}{\tilde{N}_k} \int_{\mathbf{E}_{\tau_{k+1}} \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}} f(\mathbf{z}_{\tau_{k+1}}) V_k(d\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j) \\
&\simeq \sum_{j=1}^{\tilde{N}_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} f(\mathbf{a}_{\tau_{k+1}}^{k,j}). \tag{42}
\end{aligned}$$

### 5.2. Modify the propagation of clusters

In order to diversify the simulated trajectories, and to increase the precision of the estimation, we propose to modify 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  $k^{\text{th}}$  selected sample, and  $N_{k+1}$  the size of the  $k^{\text{th}}$  propagated sample, with the convention  $N_0 = N$ . We stressed out, in section 5.1, that the propagation step aims at providing an estimation of  $\eta_{k+1} = \tilde{\eta}_k V_k$  using the selected sample  $(\tilde{\mathbf{Z}}_k^j, \tilde{W}_k^j)_{j \leq \tilde{N}_k}$ . 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  $\tilde{\eta}_k^{\tilde{N}_k} V_k$  defined by:

$$\tilde{\eta}_k^{\tilde{N}_k} V_k(f) = \sum_{j=1}^{\tilde{N}_k} \tilde{W}_k^j \delta_{\tilde{\mathbf{z}}_k^j} V_k(f) = \sum_{j=1}^{\tilde{N}_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} \delta_{\mathbf{z}_k^j} V_k(f), \tag{43}$$

where  $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$ . We denote by  $\bar{V}_k$  the Markovian kernel from  $\mathbf{E}_{\tau_k}$  to  $\mathbf{E}_{\tau_{k+1}}$  such that, for any trajectory  $\mathbf{Z}_{\tau_k}^j$ ,  $\bar{V}_k(\cdot | \mathbf{Z}_{\tau_k}^j)$  is the conditioning of  $V_k(\cdot | \mathbf{Z}_{\tau_k}^j)$  to  $\mathbf{E}_k \setminus \{\mathbf{a}_{\tau_{k+1}}^{k,j}\}$ .  $\bar{V}_k$ 's density verifies:

$$\bar{V}_k(\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j) = \frac{V_k(\mathbf{z}_{\tau_{k+1}} | \mathbf{Z}_{\tau_k}^j)}{1 - V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)} \mathbb{1}_{\mathbf{z}_{\tau_{k+1}} \neq \mathbf{a}_{\tau_{k+1}}^{k,j}}. \tag{44}$$

Using (43) we can decompose  $\tilde{\eta}_k^{\tilde{N}_k} V_k$  as follows:

$$\tilde{\eta}_k^{\tilde{N}_k} V_k(f) = \sum_{j=1}^{\tilde{N}_k} \frac{\tilde{N}_k^j}{\tilde{N}_k} \left[ V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) f(\mathbf{a}_{\tau_{k+1}}^{k,j}) + \left(1 - V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)\right) \delta_{\mathbf{z}_k^j} \bar{V}_k(f) \right]. \tag{45}$$

In the original IPS algorithm, the sample approximating  $\tilde{\eta}_k^{\tilde{N}_k} V_k$  is built by directly extending each trajectory in the selected sample. When we extend the replicates of a cluster, in average a proportion  $V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)$  of the replicates are extended in  $\mathbf{a}_{\tau_{k+1}}^{k,j}$ . This proportion of trajectories extended in  $\mathbf{a}_{\tau_{k+1}}^{k,j}$  then serves as an estimation of  $V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)$ . But it is not necessary to misspend all these replicates to estimate the probability of the preponderant

trajectory. If we use equation (45), 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  $Z_{\tau_k}^j$  until time  $\tau_{k+1}$ , 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  $\mathbf{Z}_k^j$ ,  $N_k^j$  times, where  $N_k^j = \tilde{N}_k^j + \mathbb{1}_{\tilde{N}_k^j > 0}$ . We denote  $j_i$  the index of the  $i^{\text{th}}$  replicates of  $\mathbf{Z}_k^j$ , and consider the added replicate has index 0 such that for  $i \in \{0, \dots, \tilde{N}_k^j\}$  we have  $\tilde{\mathbf{Z}}_{\tau_k}^{j_i} = \mathbf{Z}_k^j$ . 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  $W_{k+1}^{j_0} = V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) \frac{\tilde{N}_k^j}{N_k}$ , 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 1<sup>st</sup> term in the right hand side of equation (41)). 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  $\bar{V}_k(\cdot | \mathbf{Z}_{\tau_k}^j)$  and set  $W_{k+1}^{j_i} = \frac{1 - V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) \frac{\tilde{N}_k^j}{N_k}}{\tilde{N}_k^j} \frac{\tilde{N}_k^j}{N_k}$ .

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  $V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)$  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 [16], 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 section 6.

The target distributions  $\tilde{\eta}_k$  are still estimated with  $\tilde{\eta}_k^{\tilde{N}_k}$ , using equation (39), but for  $k = 0$  to  $n - 1$ ,  $\eta_{k+1}$ , the propagation of a target distribution, is now estimated by :

$$\begin{aligned} \eta_{k+1}^{N_{k+1}} &= \sum_{i=1}^{N_{k+1}} W_{k+1}^i \delta_{\mathbf{z}_{k+1}^i} = \sum_{j=1, \tilde{N}_k^j > 0}^{N_k} \sum_{i=0}^{\tilde{N}_k^j} W_{k+1}^{j_i} \delta_{\mathbf{z}_{k+1}^{j_i}} \\ &= \sum_{j=1, \tilde{N}_k^j > 0}^{N_k} \frac{\tilde{N}_k^j}{N_k} \left[ V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j) \delta_{\mathbf{a}_{\tau_{k+1}}^{k,j}} + \frac{\left(1 - V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)\right)}{\tilde{N}_k^j} \sum_{i=1}^{\tilde{N}_k^j} \delta_{\mathbf{z}_{k+1}^{j_i}} \right] \end{aligned} \quad (46)$$

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:

$$\tilde{\gamma}_k^{\tilde{\mathbf{N}}_k}(f) = \tilde{\eta}_k^{\tilde{\mathbf{N}}_k}(f) \prod_{s=0}^{k-1} \eta_s^{N_s}(G_s) \quad \text{and} \quad \gamma_k^{\mathbf{N}_k}(f) = \eta_k^{N_k}(f) \prod_{s=0}^{k-1} \eta_s^{N_s}(G_s). \quad (47)$$

In the end,  $p_h$  is estimated using the equation :

$$\hat{p}_h = \eta_n^{N_n}(f_h) \prod_{k=0}^{n-1} \eta_k^{N_k}(G_k). \quad (48)$$

The full modified version of the algorithm is presented in Table 3. We call this modified version of the IPS algorithm the IPS+M algorithm.

Throughout the rest of the paper, 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.

<p><b>Initialization :</b> <math>k = 0</math>, <math>\forall j = 1..N</math>, <math>\mathbf{Z}_0^j = (z_0)</math> and <math>W_0^j = \frac{1}{N}</math>, and <math>\tilde{W}_0^j = \frac{G_0(\mathbf{Z}_0^j)}{\sum_s G_0(\mathbf{Z}_0^s)}</math></p> <p><b>while</b> <math>k &lt; n</math> <b>do</b></p> <p style="padding-left: 20px;"><b>Selection:</b></p> <p style="padding-left: 40px;"><math>\tilde{N}_k = N</math>, and sample <math>(\tilde{N}_k^j)_{j=1..N_k} \sim Mult(N, (\tilde{W}_k^j)_{j=1..N_k})</math></p> <p style="padding-left: 40px;"><math>\forall i = 1..N_k</math>, <math>\tilde{W}_k^i := \frac{1}{N}</math></p> <p style="padding-left: 40px;"><math>\forall i = 1..N_k</math>, set <math>N_k^i = \tilde{N}_k^i + \mathbb{1}_{\tilde{N}_k^i &gt; 0}</math></p> <p style="padding-left: 20px;">set <math>N_k := \sum_{i=1}^{N_k} N_k^i</math> <b>Propagation :</b></p> <p style="padding-left: 20px;"><b>for</b> <math>j := 1..N_k</math> <b>do</b></p> <p style="padding-left: 40px;"><b>if</b> <math>N_k^j &gt; 0</math> <b>then</b></p> <p style="padding-left: 60px;">set <math>\mathbf{Z}_{\tau_{k+1}}^{j_0} = \mathbf{a}_{\tau_{k+1}}^{k,j}</math> and <math>W_{k+1}^{j_0} = V_k(\mathbf{a}_{\tau_{k+1}}^{k,j}   \mathbf{Z}_{\tau_k}^j) \sum_{i=1}^{\tilde{N}_k^j} \tilde{W}_k^{j_i}</math></p> <p style="padding-left: 60px;"><b>for</b> <math>j = 1..N_k^j</math> <b>do</b></p> <p style="padding-left: 80px;">Sample <math>\mathbf{Z}_{\tau_{k+1}}^{j_i}</math> from <math>\bar{V}_k(\cdot   \mathbf{Z}_{\tau_k}^j)</math> and set <math>W_{k+1}^{j_i} = \frac{(1 - V_k(\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}</math></p> <p style="padding-left: 20px;"><b>for</b> <math>i := 1..N_k</math> <b>do</b></p> <p style="padding-left: 40px;"><math>\tilde{W}_{k+1}^i = \frac{W_{k+1}^{j_i} G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^{j_i})}{\sum_j W_{k+1}^j G_{k+1}(\mathbf{Z}_{\tau_{k+1}}^j)}</math></p> <p style="padding-left: 20px;"><b>if</b> <math>\forall j</math>, <math>\tilde{W}_{k+1}^j = 0</math> <b>then</b></p> <p style="padding-left: 40px;"><math>\forall q &gt; k</math>, set <math>\eta_q^{N_q} = \tilde{\eta}_q^{N_q} = 0</math> and Stop</p> <p style="padding-left: 20px;"><b>else</b></p> <p style="padding-left: 40px;"><math>k := k + 1</math></p>
---

Table 3: IPS+M algorithm

### 5.3. Convergence properties of the IPS+M estimators

In this section we show that the estimator  $\hat{p}_h$  obtained with the IPS+M method have the same basic properties as the IPS estimator. With the IPS+M method,  $\hat{p}_h$  converges almost surely to  $p_h$ , it is unbiased, and it satisfies a CLT. The proofs that we provide in this section follow the reasoning of the proofs in [8]. 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.

### 5.3.1. The martingale decomposition of the anticipated biases

With the IPS+M method, we assumed that  $\forall k, \tilde{N}_k = N$ . 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 adapted to 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 an odd number  $p = 2k + 1$ ,  $\mathcal{F}_p$  is adapted to 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  $f \in \mathcal{M}(\mathbf{E}_{\tau_n})$  we let  $\Gamma_{p,2n}^{\mathbf{N}}(h)$  be define by

$$\begin{aligned} \Gamma_{2k,2n}^{\mathbf{N}}(f) &= \gamma_k^{\mathbf{N}_k}(Q_{k,n}(f)) - \gamma_k(Q_{k,n}(f)) \\ &= \gamma_k^{\mathbf{N}_k}(Q_{k,n}(f)) - \gamma_n(f) \end{aligned} \quad (49)$$

and

$$\begin{aligned} \Gamma_{2k+1,2n}^{\mathbf{N}}(f) &= \tilde{\gamma}_k^{\tilde{\mathbf{N}}_k}(V_k Q_{k+1,n}(f)) - \tilde{\gamma}_k(V_k Q_{k+1,n}(f)) \\ &= \tilde{\gamma}_k^{\tilde{\mathbf{N}}_k}(V_k Q_{k+1,n}(f)) - \gamma_n(f). \end{aligned} \quad (50)$$

Using a telescopic argument we get

$$\begin{aligned} \Gamma_{p,2n}^{\mathbf{N}}(f) &= \sum_{k=0}^{\lfloor \frac{p}{2} \rfloor} \gamma_k^{\mathbf{N}_k}(Q_{k,n}(f)) - \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(V_{k-1} Q_{k,n}(f)) \\ &\quad + \mathbb{1}_{p>0} \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(V_{k-1} Q_{k,n}(f)) - \gamma_{k-1}^{\mathbf{N}_{k-1}}(Q_{k-1,n}(f)), \end{aligned} \quad (51)$$

with the convention for  $k = 0$ ,  $\tilde{\gamma}_{-1}^{\tilde{\mathbf{N}}_{-1}}(V_{-1} Q_{0,n}(f)) = \gamma_n(f)$ .

Noticing that  $\gamma_k^{\mathbf{N}_k}(1) = \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(1) = \gamma_{k-1}^{\mathbf{N}_{k-1}}(G_{k-1})$ , we can rewrite (51) as

$$\begin{aligned} \Gamma_{p,2n}^{\mathbf{N}}(f) &= \sum_{k=0}^{\lfloor \frac{2p}{2} \rfloor} \gamma_k^{\mathbf{N}_k}(1) \left( \eta_k^{\mathbf{N}_k}(Q_{k,n}(f)) - \tilde{\eta}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(V_{k-1} Q_{k,n}(f)) \right) \\ &\quad + \mathbb{1}_{p>0} \sum_{k=1}^{\lfloor \frac{2p+1}{2} \rfloor} \tilde{\gamma}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(1) \left( \tilde{\eta}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(V_{k-1} Q_{k,n}(f)) - \Psi_{k-1}(\eta_{k-1}^{\mathbf{N}_{k-1}}(V_{k-1} Q_{k,n}(f))) \right), \end{aligned} \quad (52)$$

where for  $k = 0$ , we use the convention  $\gamma_0^{\mathbf{N}_0}(1) \tilde{\eta}_{-1}^{\tilde{\mathbf{N}}_{-1}}(V_{-1} Q_{0,n}(f)) = \gamma_n(f)$ . 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 (41) we easily get that for any  $f \in \mathcal{M}(\mathbf{E}_{\tau_{k+1}})$ :

$$\mathbb{E}_M \left[ \eta_k^{\mathbf{N}_k}(f) \mid \mathcal{F}_{2k-1} \right] = \tilde{\eta}_{k-1}^{\tilde{\mathbf{N}}_{k-1}}(V_{k-1}(f)). \quad (53)$$

For the selection steps, as the resampling schemes are the same ones as for the IPS algorithm, we still have for any  $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$ :

$$\mathbb{E}_M \left[ \tilde{\eta}_k^{\tilde{N}_k}(f) | \mathcal{F}_{2k} \right] = \Psi_k(\eta_k^N)(f). \quad (54)$$

Thus, each selection step and propagation step is conditionally unbiased. Note that  $\gamma_k^{N_k}(1)$  is  $\mathcal{F}_{2k-1}$ -measurable and  $\gamma_k^{\tilde{N}_k}(1)$  is  $\mathcal{F}_{2k}$ -measurable, so, when the samples are generated with the IPS+M algorithm,  $(\Gamma_{p,2n}^N(h))_{p \leq 2n}$  is a  $\mathcal{F}_p$ -martingale. Therefore,  $\hat{p}_h$  stays unbiased with the IPS+M method, because

$$\mathbb{E}_M[\Gamma_{2n,2n}^N(f_h)] = \mathbb{E}_M[\hat{p}_h - p_h] = 0.$$

### 5.3.2. Almost sure convergence

Thanks to this martingale decomposition, we can use the same arguments as in the proof in the chapter 7 in [8]. The hypotheses of Theorems 7.4.2 and 7.4.3 [8, page 239 and 241] are satisfied with the IPS+M method too, which yields the following theorem:

**Theorem 4.** *For any  $h \in \mathcal{M}(\mathbf{E}_{\tau_n})$ ,  $\hat{p}_h$  converges almost surely to  $p_h$ , and, for any  $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$ ,  $\eta_k^{N_k}(f)$  converges to  $\eta_k(f)$  almost surely,  $\gamma_k^{N_k}(f)$  converges to  $\gamma_k(f)$  almost surely.*

### 5.3.3. A Central Limit Theorem

**Theorem 5.** *If the potential functions verify (G) and the samples are generated with the IPS+M algorithm, then we have the following convergence in distribution:*

$$\sqrt{N}(\hat{p}_h - p_h) \xrightarrow{N \rightarrow \infty} \mathcal{N}(0, \sigma_{M,G}^2),$$

where

$$\begin{aligned} \sigma_{M,G}^2 &= \eta_0 \left( \left[ Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left( \left( 1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{V}_{k-1} \left[ Q_{k,n}(f_h) - \bar{V}_{k-1} Q_{k,n}(f_h) \right]^2 \right) \\ &+ \sum_{k=1}^n \tilde{\gamma}_k(1)^2 \tilde{\eta}_{k-1} \left( \left[ V_{k-1} Q_{k,n}(f_h) - \tilde{\eta}_{k-1} V_{k-1} Q_{k,n}(f_h) \right]^2 \right). \end{aligned} \quad (55)$$

*Proof.* This proof is very similar to what is done in Chapter 9 of [8]. In order to prove that  $\hat{p}_h$  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  $p^{\text{th}}$  selection and propagation by:

$$\begin{aligned} M_{p,2n}^N(f) &= \sum_{k=0}^{\lfloor \frac{p}{2} \rfloor} \eta_k^{N_k}(f_{2k}) - \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}} V_{k-1}(f_{2k}) \\ &+ \mathbb{1}_{p>0} \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\eta}_{k-1}^{\tilde{N}_{k-1}}(f_{2k-1}) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}})(f_{2k-1}). \end{aligned} \quad (56)$$

For  $j \in \{1, \dots, N\}$  we let

$$U_{(2k+1)N+j}^N(f) = \frac{1}{\sqrt{N}} \left( f_{2k+1}(\tilde{\mathbf{Z}}_k^j) - \Psi_k(\eta_k^{N_k})(f_{2k+1}) \right). \quad (57)$$

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

$$U_{2kN+s}^N(f) = \frac{1 - V_k(\mathbf{a}_{\tau_{k+1}}^{k,j} | \mathbf{Z}_{\tau_k}^j)}{\sqrt{N}} \left( f_{2(k+1)}(\mathbf{Z}_{k+1}^{j_i}) - \bar{V}_k(f_{2(k+1)})(\mathbf{Z}_k^j) \right). \quad (58)$$

For  $j \in \{1, \dots, N_0\}$ , let

$$U_j^N(f) = \frac{1}{\sqrt{N}} \left( f_0(\mathbf{Z}_0^j) - \eta_0(f_0) \right). \quad (59)$$

Thus,

$$\sqrt{N} M_{p,2n}^N(f) = \sum_{k=0}^{(p+1)N} U_k^N(f). \quad (60)$$

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

$$\begin{aligned} \langle \sqrt{N} M_{p,2n}^N(f) \rangle_p &= \sum_{k=0}^{(p+1)N} \mathbb{E}[U_k^N(f)^2 | \mathcal{P}_{k-1}^N] \\ &= \eta_0^N \left( [f_0 - \eta_0^N(f_0)]^2 \right) \\ &\quad + \sum_{k=1}^{\lfloor \frac{2p}{2} \rfloor} \tilde{\eta}_{k-1}^N \left( (1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}))^2 \bar{V}_{k-1} [f_{2k} - \bar{V}_{k-1} f_{2k}]^2 \right) \\ &\quad + \sum_{k=1}^{\lfloor \frac{2p+1}{2} \rfloor} \tilde{\eta}_{k-1}^N \left( [f_{2k-1}(\mathbf{Z}_{\tau_{k-1}}) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}}) f_{2k-1}]^2 \right). \end{aligned} \quad (61)$$

As  $\eta_k^{N_k}$  and  $\tilde{\eta}_k^N$  converge almost surely to  $\eta_k$  and  $\tilde{\eta}_k$ ,  $\langle \sqrt{N} M_{p,2n}^N(f) \rangle_n$  converge in probability

to

$$\begin{aligned}
\sigma_p^2(f) &= \eta_0 \left( \left[ f_0 - \eta_0(f_0) \right]^2 \right) \\
&+ \sum_{k=1}^{\lfloor \frac{p-1}{2} \rfloor} \tilde{\eta}_{k-1} \left( \left( 1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{V}_{k-1} \left[ f_{2k} - \bar{V}_{k-1} f_{2k} \right]^2 \right) \\
&+ \sum_{k=1}^{\lfloor \frac{p+1}{2} \rfloor} \tilde{\eta}_{k-1} \left( \left[ f_{2k-1}(\mathbf{Z}_{\tau_{k-1}}) - \tilde{\eta}_{k-1} f_{2k-1} \right]^2 \right).
\end{aligned} \tag{62}$$

By application of the Lindeberg's theorem for triangular array (see for instance Theorem 4 on page 543 in [18]), 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

$$\begin{aligned}
&\sqrt{N} \left( \eta_k^{N_k} Q_{k,n}(f_h) - \tilde{\eta}_{k-1}^N V_{k-1} Q_{k,n}(f_h) \right) \\
&\xrightarrow{N \rightarrow \infty} \tilde{\eta}_{k-1} \left( \left( 1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{V}_{k-1} \left[ Q_{k,n}(f_h) - \bar{V}_{k-1} Q_{k,n}(f_h) \right]^2 \right)
\end{aligned}$$

and if for  $p \neq 2k - 1$  we take  $f_p = 0$  and for  $p = 2k - 1$   $f_{2k-1} = V_{k-1} Q_{k,n}(f_h)$ , we get that

$$\begin{aligned}
&\sqrt{N} \left( \tilde{\eta}_{k-1}^N (V_{k-1} Q_{k,n}(f_h)) - \Psi_{k-1}(\eta_{k-1}^{N_{k-1}})(V_{k-1} Q_{k,n}(f_h)) \right) \\
&\xrightarrow{N \rightarrow \infty} \tilde{\eta}_{k-1} \left( \left[ V_{k-1} Q_{k,n}(f_h) - \tilde{\eta}_{k-1} V_{k-1} Q_{k,n}(f_h) \right]^2 \right).
\end{aligned}$$

Following from Theorem 4,  $\gamma_k^{N_k}(1)^2$  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  $\sqrt{N}\Gamma_{2N,2n}^{\mathbf{N}}(f_h)$  converges in law to a centered Gaussian with variance

$$\begin{aligned}
\sigma_{M,G}^2 &= \gamma_0(1)^2 \eta_0 \left( \left[ Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right) \\
&+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left( \left( 1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) \right)^2 \bar{V}_{k-1} \left[ Q_{k,n}(f_h) - \bar{V}_{k-1} Q_{k,n}(f_h) \right]^2 \right) \\
&+ \sum_{k=1}^n \tilde{\gamma}_{k-1}(1)^2 \tilde{\eta}_{k-1} \left( \left[ V_{k-1} Q_{k,n}(f_h) - \tilde{\eta}_{k-1} V_{k-1} Q_{k,n}(f_h) \right]^2 \right).
\end{aligned} \tag{63}$$

□



#### 5.3.4. Variance reduction

**Theorem 6.** *The variance of the original IPS can be decomposed as follows:*

$$\sigma_{IPS,G}^2 = \sigma_{M,G}^2 + \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} \left( v_k(\mathbf{Z}_{\tau_{k-1}}) \bar{V}_{k-1} \left( \left[ Q_{k,n}(f_h)(\mathbf{a}_{\tau_k}) - Q_{k,n}(f_h)(\mathbf{Z}_{\tau_k}) \right]^2 \right) \right), \quad (64)$$

where  $v_k(\mathbf{Z}_{\tau_{k-1}}) = V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) (1 - V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}))$ . Therefore we have  $\sigma_{M,G}^2 \leq \sigma_{IPS,G}^2$ .

*Proof.*

$$\sigma_{IPS,G}^2 = \sum_{k=0}^n \gamma_k(1)^2 \eta_k \left( \left[ Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h) \right]^2 \right) \quad (65)$$

$$= \eta_0 \left( \left[ Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right)$$

$$+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} V_{k-1} \left( \left[ Q_{k,n}(f_h) - V_{k-1} Q_{k,n}(f_h) + V_{k-1} Q_{k,n}(f_h) - \eta_k Q_{k,n}(f_h) \right]^2 \right)$$

$$= \eta_0 \left( \left[ Q_{0,n}(f_h) - \eta_0 Q_{0,n}(f_h) \right]^2 \right)$$

$$+ \sum_{k=1}^n \gamma_k(1)^2 \tilde{\eta}_{k-1} V_{k-1} \left( \left[ Q_{k,n}(f_h) - V_{k-1} Q_{k,n}(f_h) \right]^2 \right)$$

$$+ \sum_{k=1}^n \tilde{\gamma}_{k-1}(1)^2 \tilde{\eta}_{k-1} \left( \left[ V_{k-1} Q_{k,n}(f_h) - \tilde{\eta}_{k-1} V_{k-1} Q_{k,n}(f_h) \right]^2 \right) \quad (66)$$

Temporarily using the notation  $V_{k-1}(\mathbf{a}_{\tau_k} | \mathbf{Z}_{\tau_{k-1}}) = p_k$ , for any  $f \in \mathcal{M}(\mathbf{E}_{\tau_k})$ , we get

$$\begin{aligned} & V_{k-1} \left( \left[ f(\mathbf{Z}_{\tau_k}) - V_{k-1} f \right]^2 \right) \\ &= V_{k-1} \left( \left[ f(\mathbf{Z}_{\tau_k}) - p_k f(\mathbf{a}_{\tau_k}) - (1-p_k) \bar{V}_{k-1}(f) \right]^2 \right) \\ &= V_{k-1} \left( f(\mathbf{Z}_{\tau_k})^2 - 2p_k f(\mathbf{Z}_{\tau_k}) f(\mathbf{a}_{\tau_k}) - 2(1-p_k) f(\mathbf{Z}_{\tau_k}) \bar{V}_{k-1} f \right. \\ &\quad \left. + p_k^2 f(\mathbf{a}_{\tau_k})^2 + 2p_k f(\mathbf{a}_{\tau_k}) \bar{V}_{k-1} f + (1-p_k)^2 (\bar{V}_{k-1} f)^2 \right) \\ &= p_k f(\mathbf{a}_{\tau_k})^2 + (1-p_k) \bar{V}_{k-1}(f^2) \\ &\quad - 2p_k^2 f(\mathbf{a}_{\tau_k})^2 - 2p_k(1-p_k) f(\mathbf{a}_{\tau_k}) \bar{V}_{k-1}(f) \\ &\quad - 2p_k(1-p_k) f(\mathbf{a}_{\tau_k}) \bar{V}_{k-1} f - 2(1-p_k) f^2 \bar{V}_{k-1}(f)^2 \\ &\quad + p_k^2 f(\mathbf{a}_{\tau_k})^2 + 2p_k f(\mathbf{a}_{\tau_k}) \bar{V}_{k-1} f + (1-p_k)^2 \bar{V}_{k-1}(f)^2 \\ &= p_k(1-p_k) \left[ f(\mathbf{a}_{\tau_k})^2 - 2f(\mathbf{a}_{\tau_k}) \bar{V}_{k-1} f + \bar{V}_{k-1}(f^2) \right] \\ &\quad + (1-p_k)^2 \left( \bar{V}_{k-1}(f^2) - \bar{V}_{k-1}(f)^2 \right) \\ &= p_k(1-p_k) \bar{V}_{k-1} \left( \left[ f(\mathbf{a}_{\tau_k}) - f(\mathbf{Z}_{\tau_k}) \right]^2 \right) + (1-p_k)^2 \left( \bar{V}_{k-1}(f^2) - \bar{V}_{k-1}(f)^2 \right). \quad (67) \end{aligned}$$

In particular, for  $f = Q_{k,n}(f_h)$  we get

$$\begin{aligned} & V_{k-1} \left( \left[ Q_{k,n}(f_h) - V_{k-1} Q_{k,n}(f_h) \right]^2 \right) \\ &= p_k(1-p_k) \bar{V}_{k-1} \left( \left[ Q_{k,n}(f_h)(\mathbf{a}_{\tau_k}) - Q_{k,n}(f_h) \right]^2 \right) + (1-p_k)^2 \left( \bar{V}_{k-1}(f^2) - \bar{V}_{k-1}(f)^2 \right) \end{aligned} \quad (68)$$

Plugging (68) into the second line of (66), yields (64).  $\square$

## 6. Efficient extension of the trajectories using the Memorization method

This section presents the memorization method that was first introduced in [15]. Remember that we considered 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  $\mathbf{a}_t$ .

### 6.1. 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.

### 6.2. Remarks on improving the IPS+M

Note that the IPS+M, greatly unbalances the weights of the propagated samples. Consequently it is useless to consider an algorithm which triggers a resampling step, when the value of on the effective sample size is below a threshold. Indeed, the weights are so unbalanced that the effective sample size would always be very low and would trigger a resampling at each step.

### 6.3. Principle of the memorization method

#### 6.3.1. Work with the differentiation time

The key idea of the memorization method is to consider the stopping time  $\tau$  defined such that:

$$\forall s < \tau, \quad Z_s = a_s \quad \text{and} \quad Z_\tau \neq a_\tau. \quad (69)$$

This time  $\tau$  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  $\tau$  knowing  $\tau \leq t$ , and set  $\mathbf{Z}_{\tau^-} = \mathbf{a}_{\tau^-}$ ,
2. generate  $Z_\tau$  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.

### 6.3.2. Generate $\tau$ knowing $\tau \leq t$

To achieve this first step, the authors in [15] propose to generate  $\tau$  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 cdf of  $\tau$  knowing  $\tau \leq t$ :

$$F(v) = \mathbb{P}(\tau < v | \tau \leq t), \quad (70)$$

and we denote by  $F^{-1}$  its generalized inverse defined by

$$F^{-1}(x) = \inf_{v>0} \{v \mid F(v) \geq x\}. \quad (71)$$

We also denote by  $\tilde{F}$  the function defined by

$$\tilde{F}(v) = \mathbb{P}(\mathbf{Z}_{v^-} = \mathbf{a}_{v^-}) = \prod_{k=0}^{n(\mathbf{a}_v)} \exp \left[ -\Lambda_{a_{s_k}}(t_k) \right] \prod_{k=1}^{n(\mathbf{a}_v)} \left( K_{a_{s_k}}(a_{s_k}) \right)^{\mathbb{1}_{t_k > 0}} \quad (72)$$

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 times  $s_k$  where  $K_{z_k^-}(z_k) \neq 1$ , 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

$$\tilde{F}^{-1}(x) = \sup_{v>0} \{v \mid \tilde{F}(v) \leq x\}. \quad (73)$$

$\tilde{F}^{-1}$  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  $\tau \mid \tau \leq t$  which is a truncated random variable. The simulation of such random variables is also presented in [10]. 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

$$\begin{aligned} F^{-1}(U) &= \inf_{v>0} \{v \mid F(v) \geq U\} \\ &= \sup_{v>0} \{v \mid \tilde{F}(v) \leq 1 - U(1 - p_{\mathbf{a}_t})\}, \\ &= \tilde{F}^{-1}\left(1 - U(1 - p_{\mathbf{a}_t})\right). \end{aligned} \quad (74)$$

Also, as  $U$  has uniform distribution on  $[0, 1]$ ,  $\tilde{U} = 1 - U(1 - p_{\mathbf{a}_t})$  is a uniform on  $[p_{\mathbf{a}_t}, 1]$ . Therefore, sampling with the inverse transform method is equivalent to simulating  $\tilde{U} \sim Unif(p_{\mathbf{a}_t}, 1)$  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 inversible on this interval, and  $\tilde{F}^{-1}$  corresponds to  $F$ '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

$$\forall v \in [s_k, s_{k+1}), \quad \tilde{F}(v) = \tilde{F}(s_k) \times \exp[-\Lambda_{a_{s_k}}(v - s_k)]. \quad (75)$$

So in particular, for  $v = \tilde{F}^{-1}(\tilde{U})$ , we have :

$$\tilde{U} = \tilde{F}(\tilde{F}^{-1}(\tilde{U})) = \tilde{F}(s_k) \times \exp[-\Lambda_{a_{s_k}}(\tilde{F}^{-1}(\tilde{U}) - s_k)], \quad (76)$$

or equivalently

$$\log\left(\frac{\tilde{F}(s_k)}{\tilde{U}}\right) = \int_0^{\tilde{F}^{-1}(\tilde{U}) - s_k} \lambda_{a_{s_k}}(u) du. \quad (77)$$

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  $\tau | \tau \leq t$  we proceed as follow:

1. Generate  $\tilde{U} \sim Unif(p_{\mathbf{a}_t}, 1)$ , 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 we 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 we 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 were not satisfied, set  $k = k + 1$ , if  $k \leq n(\mathbf{a}_s)$  repeat the steps 2 to 4

#### 6.4. The memorization in the IPS+M algorithm

In the IPS+M algorithm, assuming we apply the memorization method on an interval  $(\tau_{k-1}, \tau_k]$  and for the  $i^{th}$  cluster, we apply it knowing that  $\mathbf{Z}_{\tau_{k-1}} = \mathbf{a}_{k-1}^i$ . Note that this

trajectory  $\mathbf{a}_{k-1}^i$  is not necessarily preponderant but when we extend it into  $\mathbf{a}_k^i$ , the piece of trajectory  $\mathbf{a}_{(\tau_{k-1}, \tau_k]}^i$  is preponderant because :

$$\mathbb{P}(\mathbf{Z}_{(\tau_{k-1}, \tau_k]} = \mathbf{a}_{(\tau_{k-1}, \tau_k]}^i | \mathbf{Z}_{\tau_{k-1}} = \mathbf{a}_{k-1}^i) = \mathbb{P}(\mathbf{Z}_{\tau_k} = \mathbf{a}_{\tau_k}^i | \mathbf{Z}_{\tau_{k-1}} = \mathbf{a}_{k-1}^i) > 0. \quad (78)$$

So, in the IPS we try to generate trajectories of the cluster that verify  $\mathbf{Z}_{\tau_{k-1}} = \mathbf{a}_{k-1}^i$ , but avoid the piece of trajectory  $\mathbf{a}_{(\tau_{k-1}, \tau_k]}^i$ .

## 7. Numerical illustrations

In order to confirm our results numerically, we have applied the IPS method and the IPS+M method to two two-components system.

### 7.1. The heated-room system

The first 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 rule the temperature can be derived from the physics.  $x_e$  is the exterior temperature.  $\beta_1$  is the rate of the heat transition with the exterior.  $\beta_2$  is the heating power of each heater. The differential equation giving the evolution of the 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 that happen outside boundaries, 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 3. The probability of failure  $p$  was estimated to  $2.71 \times 10^{-5}$  thanks to a massive Monte-Carlo of  $10^7$  simulations.

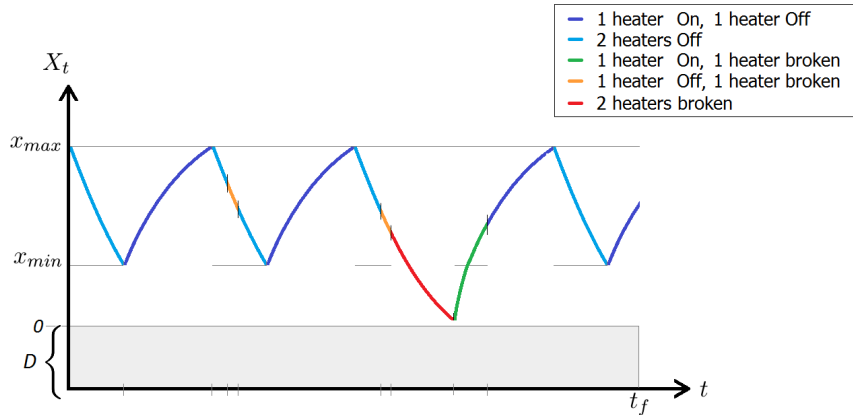


Figure 3: A possible trajectory of the heated-room system  
(the mode is represented with colors)

### 7.2. Results of the simulations for the heated-room system

The results of the simulation study for the heated-room system are displayed in Table 4. Here we have used the potential functions proposed in section 4. The value of  $\alpha$  was set to 1.1. We have tried different values of  $\alpha$  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 times slower than Monte-carlo method. So in terms of computational cost the IPS+M is slightly more efficient than the Monte-Carlo Method.

### 7.3. Remark on the SMC with Memorization

We have seen that it is possible to improve the IPS method to make it similar to the SMC method. We may, therefore, think that the IPS+M algorithm could be improved by adding adaptive optional re-sampling steps in order to get a SMC+M algorithm. In practice, however, it is not beneficial to add these adaptive optional re-sampling steps. Indeed we have noticed that, as we greatly modify the propagation process, the weights are greatly

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

Table 4: Empirical means and empirical variances on 100 runs with  $N = 10^5$  for the MC, the IPS and the IPS+M methods

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.

#### 7.4. 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.

#### 7.5. Results of the simulations for the dam system

The results of the simulation study for the dam system are displayed in Table 5. Here we have used the potential functions:

$$\forall k < n, \quad G(\mathbf{Z}_{\tau_k}) = \exp [\alpha_1(x_{lim} - X_{\tau_k}) + \alpha_2(b(Z_{\tau_k}) + 1)^2]. \quad (79)$$

The value of  $\alpha_1$  was set to  $-0.9$  and the value of  $\alpha_2$  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 5. The results highlight that the IPS 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

		MC	IPS	IPS+M
	$\hat{p}$	$1.12 \times 10^{-4}$		
	$\hat{\sigma}^2$	$1.12 \times 10^{-9}$		
$n = 5$	$\hat{p}$		$1.75 \times 10^{-4}$	$1.12 \times 10^{-4}$
	$\hat{\sigma}^2$		$3.08 \times 10^{-8}$	$4.37 \times 10^{-9}$

Table 5: Empirical means and empirical variances on 50 runs with  $N = 10^5$  for the MC, the IPS and the IPS+M methods

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.

## 8. Conclusion

This paper investigates the application of the IPS method to PDMPs. 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 TCL. 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 [17] for the IPS method in order to get an estimator of the variance for the IPS+M estimator.



## References

- [1] H. Chraïbi, A. Dutfoy, T. Galtier, and J. Garnier. Optimal input potential functions in the interacting particle system method. *Archive ouverte HAL*, (hal-01922264), 2018.
- [2] H. Chraïbi, J.-C. Houbedine, and A. Sibling. Pycatshoo: Toward a new platform dedicated to dynamic reliability assessments of hybrid systems. PSAM congress, 2016.
- [3] B. Cloez, R. Dessalles, A. Genadot, F. Malrieu, A. Marguet, and R. Yvinec. Probabilistic and piecewise deterministic models in biology. *ESAIM: Proceedings and Surveys*, 60:225–245, 2017.
- [4] 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)*, 46(3):353–388, 1984.
- [5] M.H.A. Davis. *Markov Models and Optimization*. Chapman and Hall, Boca Raton, 1993.
- [6] B. de Saporta, F. Dufour, and H. Zhang. *Numerical Methods for Simulation and Optimization of Piecewise Deterministic Markov Processes*. Wiley, Hoboken, 2015.
- [7] B. De Saporta, F. Dufour, and H. Zhang. *Numerical methods for simulation and optimization of piecewise deterministic Markov processes: application to reliability*. John Wiley & Sons, 2015.
- [8] P. Del Moral. *Feynman-Kac Formulae, Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.
- [9] P. Del Moral and J. Garnier. Genealogical particle analysis of rare events. *The Annals of Applied Probability*, 15(4):2496–2534, 2005.
- [10] L. Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265. ACM, 1986.
- [11] 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*, pages 64–69. IEEE, 2005.
- [12] P. Fearnhead and P. Clifford. On-line inference for hidden markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899, 2003.
- [13] M. Gerber, N. Chopin, and N. Whiteley. Negative association, ordering and convergence of resampling methods. *arXiv preprint arXiv:1707.01845*, 2017.
- [14] J.D. Hol, T.B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006.
- [15] 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.
- [16] P.-E. Labeau. Probabilistic dynamics: estimation of generalized unreliability through efficient Monte-Carlo simulation. *Annals of Nuclear Energy*, 23(17):1355–1369, 1996.
- [17] A. Lee and N. Whiteley. Variance estimation in the particle filter. *Biometrika*, 105(3):609–625, 2018.
- [18] A.N. Shiryaev. *Probability, second edition*. Springer-Verlag, volume 95 in graduate texts in mathematics., New York, 1996.
- [19] C. Vergé. *Modèle d’îlots de particules et application en fiabilité*. PhD thesis, Ecole Polytechnique, 2015.
- [20] N. Whiteley, A.M. Johansen, and S. Godsill. Monte Carlo filtering of piecewise deterministic processes. *Journal of Computational and Graphical Statistics*, 20(1):119–139, 2011.