The cell-to-boundary method in the frame of memorization-based Monte Carlo algorithms. A new computational improvement in dynamic reliability

P.E. Labeau¹,a,*, E. Ziob

¹Service de Métrologie Nucléaire (CP165), Université Libre de Bruxelles, Avenue FD Roosevelt, 50, 1050 Bruxelles, Belgium
bDipartimento di Ingegneria Nucleare, Politecnico di Milano, Via Ponzio 34/3, 20133 Milano, Italy

Abstract

Dynamic reliability aims at estimating failure risks associated with very rare scenarios, while accounting for the system dynamic evolution. In a Monte Carlo game devised for this purpose, the most time consuming operation consists in performing these dynamic calculations, which are repeated in thousands of histories. In order to save computer resources, the idea of memorizing information on the dynamic trajectories before the simulation was investigated. Two approaches were propounded: the cell-to-boundary (CTB) method, and algorithms based on the memorization of the most probable evolution (MPE) from each initial state. This paper presents a way to combine both methods, in order to further reduce the numerical workload of the simulation. A memorization of second-order MPEs is also propounded, to better investigate transients following the failure of a control means. These techniques are illustrated on the previously defined application of a PWR pressurizer. © 1998 IMACS/Elsevier Science B.V.

Keywords: Dynamic reliability; Monte Carlo; Memorization of dynamic-evolutions

1. Introduction

Conventional reliability techniques (e.g. fault trees and event trees) perform a risk assessment on a logical and static basis, and were not developed to give allowance to the interaction between the dynamic evolution of a system in the propagation of an accident and the stochastic changes in its hardware configuration [1–3]. As dynamic approaches to risk analysis have been developed, Monte Carlo simulation has turned out to be the most powerful way to take up the numerical challenge of such a treatment, while allowing for flexibility in the assumptions and for sensitivity analysis.

An analog Monte Carlo algorithm can indeed be easily deduced from the description of the system evolution: following an initiating event, the system starts evolving in one of its possible initial hardware
configurations according to the dynamics in this state; when a transition to another state takes place, due to either a stochastic change in the state of a component or the action of an automatic or human-driven control means, it entails a bifurcation in the trajectory followed in the process variables space by the system, since it must now obey the set of evolution equations corresponding to the new configuration. If \( n \) process variables are considered and \( \alpha \) different configurations are possible, the problem is thus defined in a phase space made of \( \alpha \) replicas of \( \mathbb{R}^n \). A random walk is defined by sampling the distribution of the next time of transition out of the current state, by performing the associated dynamic calculations up to the transition and by sampling the new configuration from the transition probabilities. This process is pursued until a maximal time, defining the accident duration, is reached or until a system failure occurs. Two failure modes are considered: the system enters one of the predefined unacceptable configurations, or the stochastic changes in the system dynamic evolution bring it out of a safety domain in the process variables space. Any further details on the concepts of dynamic reliability as well as a review of Monte Carlo simulation in this field can be found in our companion paper [4].

As explained in this first paper, a Monte Carlo game devised on this basis, though simple, turns out to be inefficient in practice for an accurate estimation of low risks – which are of interest for the analyst however – in acceptable computer times.

Indeed, the system unreliability as a function of time can be obtained by using a binary estimator, scoring a unit contribution for all times greater than the failure time in a given history. But failures are unlikely to take place in a reliable engineered installation and very large numbers of histories have to be played in order to reduce the statistical uncertainty on the estimates of unreliability. This situation is made worse because the time consuming computation of the system dynamic evolution has to be repeated alongside each history. Therefore, the memorization of trajectories in the process variables space during a pre-simulation phase has been thought as a way to reduce the corresponding numerical burden. This idea has been used in two different ways. The CTB method consists in defining in phase space a grid of nodes, from which dynamic calculations in all possible configurations are performed up to a control means or an exit out of a safety domain in phase space; this information is then used through an interpolation scheme during the simulation. On the other hand, the memorization stage of the MPE-based algorithms focuses on the trajectories which are likely to be (very often) followed on a large part of the accident duration the Monte Carlo runs.

Section 2 presents a more detailed description of both methods, compares their respective strengths and weaknesses, and propounds from these a way to combine them, when the first unexpected event to occur during a history is a stochastic transition out of an MPE. Section 3 is then dedicated to the technical features of this hybrid method, and to the implementation of a memorization of second-order MPEs, which is somehow similar to the combined technique in case of uncorrect working of a control means alongside a (first-order) MPE. These new algorithms have been tested on the application of a PWR pressurizer and their performances are compared and discussed in Section 4. Some concluding remarks end the paper.

2. Comparison of existing memorization techniques

Before presenting the pros and cons of both methods, let us first briefly summarize how they were developed, and what they can achieve.
The CTB method [5] addresses the intrinsic slowness – in the frame of the simulation – of the numerical solution of the physical models embodying the system dynamics. It presents a discretization approach in which the process variables space continuum is replaced with a discrete grid of nodes. In a pre-simulation phase, some quantities of interest for the simulation are computed in correspondence of all nodes. They provide the basis for an interpolation procedure suitable to describe the evolution of the point representative of the system state. The following characteristics are associated with each node, in each configuration the system can be found in:

- The time interval $t_c$ required to go from this node to a first control threshold, and the value $\tilde{x}_c$ of the process variables at the intersection with the system trajectory.
- The same quantities ($t_{\text{out}}$ and $\tilde{x}_{\text{out}}$) related to the first intersection of the dynamic trajectory of the system’s representative point with the border of the safety domain.
- In both cases, the integral of the transition rate out of the current hardware state between the node and these intersection points ($\Lambda_c$ and $\Lambda_{\text{out}}$), if the next transition time distribution is assumed exponential in each state; when the system enters a new state at this node during a history, this information is used to determine the kind of event (stochastic or dynamic transition, system failure) that will be next encountered, by comparing these memorized values to the one corresponding to a sampled transition time.

Since in many cases the next transition corresponds to the actuation of the closest control means, a large amount of dynamic calculations can be avoided during the simulation, thereby strongly decreasing the mean computer time per history.

The MPE-based technique aims at improving the simulation itself, at the same time it reduces the numerical workload of computing the dynamic trajectories [6,7]. It follows from the observation that, for a given initiating event and from a given initial situation, a reliable system has a very large probability – that can be assessed – to evolve deterministically between control devices, without undergoing any stochastic transition, until the transient that was induced is mitigated. During the pre-simulation phase, the main characteristics ($t_c, t_{\text{out}}, \ldots$ current state, probability to stay in the MPE.) of each section of these MPEs are computed from their previous intersection with a control threshold and are memorized.

But this method presents two other major features:

- In the memorization phase, the probabilities to reach the different stages of the MPEs without undergoing any unexpected event (stochastic transition or failure of a control device) can be obtained.
- The definition of an efficient unbiased estimator of unreliability is coupled with the MPE-based approach. It consists in scoring, from each deterministic section between two transitions in the random walk, the probability of a failure, either by entering an unacceptable state or by escaping the safety domain. This estimator thereby collects contributions to the score even in the absence of failure. While computing the MPEs, we calculate, after the correct working of each control device, the probability of reaching a safety border in the corresponding new configuration. We can then sum up these contributions to obtain the partial scores collected up to any stage of all MPEs.
This memorized information can be used in several ways to improve the simulation:

- The stage of an MPE where an unexpected event takes place is sampled from the memorized probabilities, and all dynamic calculations are bypassed up to this point.
- To better investigate rare scenarios, an unexpected event is forced to occur before the end of the accident duration in each history. The estimation can be kept unbiased, because the probability to follow each MPE up to the maximal time and the score associated with this situation were calculated before the simulation.
- A systematic exploration of accidental sequences is made by estimating the unreliabilities conditional to an unexpected event occurring along an MPE. The simulation then boils down to playing independently batches of histories for each kind of accidental scenarios and to weighing these estimations by their known probability of occurrence.

From these short descriptions, we can try to highlight the main advantages and drawbacks of each technique. The CTB method asks for a long preparation, with a large number of calculations if the number of states and process variables is important (e.g. 12 variables and 900 states in the application treated in Section 4). But it strongly reduces the number of calculations of dynamic evolutions during the simulation, by interpolating memorized results, whatever trajectory is followed in phase space by the system’s representative point. However, the interpolation entails an error on the process variables integration, which depends on the definition of the grid of nodes. But this error is acceptable, in comparison with the uncertainties pertaining to the values of the reliability data. The main drawback of the CTB approach is that the probability of finding the system in some rare states at given nodes can be very small and even zero. This makes useless part of the memorized information, what cuts down the technique efficiency. On the other hand, an optimized discretization scheme of the phase space could be developed, in which the mesh is kept dense around the most probable paths of evolution while it is more sparse in those areas of the phase space of rare occurrence.

The memorization of the MPEs is fast, because a very limited number of such histories have to be considered. All the memorized data are used in the simulation, both to speed it up and to define a more efficient Monte Carlo game, in which the simulation is driven towards unexpected events likely to lead to important damage of the system. Nonetheless, the MPE-based game presents the following weakness: once an unexpected event has occurred, the current history is pursued in an analog way.

Though these approaches are very different, their advantages appear complementary: the memorization of the MPEs allows to obtain accurate estimations of unreliability by simulating only risky scenarios, while the CTB approach accelerates the solution of the dynamic physical model in all configurations. Therefore, a natural combination of both techniques is obvious: on each section of an MPE between two control thresholds, a grid of nodes is defined and a CTB-like memorization can take place before investigating the accidental evolutions following a stochastic transition out of this section of the MPE. The practical details of this idea are given in the next section.

3. A combined algorithm

Before presenting this hybrid technique, let us remark that the MPE-based approach leads to independent simulations of accidental scenarios that can originate from each stage of the MPE, either
by a stochastic transition or the failure of a control means. Because these two possibilities usually correspond to probabilities of different orders of magnitude (stochastic transitions being rare in a transient) and to different numbers of possible new states (a control device often presenting only two states), separate treatments are considered.

Let us first examine scenarios initiated by a stochastic transition out of an MPE. The implementation of a CTB memorization from nodes located on each section of the MPEs raises some questions: how to determine the number of nodes? How to distribute them on the section of MPE? In which states has this memorization to be performed?

One could choose to define the grid of nodes at the same time the MPEs are calculated and memorized, e.g. by placing the nodes at constant time intervals. By doing so, a strongly irregular distribution of the grid alongside the MPEs could be obtained. Indeed, as the time intervals elapsed between the control thresholds limiting the sections can be very different, some sections would contain very few nodes, and in some cases the interpolation procedure would have to be carried on between nodes belonging to different sections. Therefore, the grid has to be defined section by section, after the memorization phase, the intersections of the MPEs with control thresholds having to be part of the grid. The intermediate nodes are placed regularly in between. We choose to do it proportionally to the value of the integral of the transition rate out of the hardware state in the current section, starting from the control threshold at the beginning of this section. Indeed, sampled values of this quantity are used to define when stochastic transitions occur along a section. This procedure then requires one step-by-step repetition of the calculation of the dynamic trajectories constituting the MPEs, but such an operation is not much time consuming.

Let \( k_i^{\text{max}} \) be the number of sections in the MPE initiated in state \( i \), and \( \Delta t_k \) the time interval elapsed in section \( k \). We introduce \( \bar{N}_{\text{nod}} \), mean number of nodes per section, such that the mean time interval \( \bar{t} \) between the nodes is

\[
\bar{t} = t_{\text{ad}} / k_i^{\text{max}} / (\bar{N}_{\text{nod}} - 1)
\]

where \( t_{\text{ad}} \) is the accident duration. The number of nodes in section \( k \) is chosen to be defined by

\[
N_{\text{nod}}^k = \text{int}(\Delta t_k / \bar{t}) + 1
\]

provided that this quantity lies in a predefined interval \([N_{\text{nod}}^{\text{min}}, N_{\text{nod}}^{\text{max}}]\). The next operation is to find which states the system can have a stochastic transition to from the hardware state in which the system lies in section \( k \), and to keep in memory the probabilities of these transitions. We have chosen to take into account all possible states for the following reason: dangerous transients with unacceptable consequences can originate from very unlikely events, thereby justifying the allowance given to them.

The next step consists in performing the CTB memorization from the nodes for all possible transitions. The characteristics kept in memory are the same as those listed in Section 2. But it has to be underlined that, in this case, \( \Lambda_{\text{out}} \) directly relates to a contribution to the score, since this integral allows to calculate the probability of exiting the safety domain before the end of the accident duration.

The last stage is concerned with the exploitation of the memorized information. A time of transition out of the MPE is sampled before the end of the current section; we can determine the two nodes surrounding the transition point, whose coordinates can be obtained by interpolation. The new state is then sampled from the transition probabilities that were memorized. Since we score the probability of a
failure in each state the system goes in during a history, even if this failure does not occur, there is no interest in actually sampling an exit out of the safety domain. Therefore, we can force a transition to take place before reaching a safety border; by doing so, we extend the duration of the history and it can improve the investigation of rare scenarios for which two consecutive transitions occur. This operation and the calculation of the corresponding statistical weight can be performed at no computer cost, on the basis of the interpolated value of $\lambda_{\text{out}}$, which defines the maximal transition time. If this biased sampling leads the system up to the closest control threshold, no supplementary dynamic calculations are required up to there, and an analog simulation can be pursued from there. The gain in computation time is more important for long sections for which more histories are played and the evolution up to the next control threshold is longer.

One has to be aware that trajectories in a given state from two neighboring nodes may be very different: only one of them could reach a given control threshold or exit the safety domain. Logical tests can be added to detect these very rare situations and drive the simulation back to the analog algorithm to avoid performing incorrect interpolations.

We now have to look at accidental situations initiated by the failure of a control system along an MPE. In this case, there is usually a very small (one or two) number of configurations the system can then evolve in. In each of these, for a well-protected installation, there is still then a large probability for it to evolve between operational control means without stochastic failures. These trajectories following a first failed control are designated as second-order most probable evolutions (SOMPEs). Their characteristics can also be memorized before simulating these uncontrolled scenarios. Once again, the partial scores accumulated along the SOMPEs are expected values, and independent simulations of the different types of accidental sequences out of these SOMPEs can be performed.

Very different results are to be expected from the CTB memorization in the case of a stochastic transition out of an MPE and from the memorization of SOMPEs after the failure of a control device along an MPE. In the first case, the main objective is a reduction of the computation time by bypassing the dynamic calculations in the first two steps of the accidental sequences, though rare scenarios can also better be detected thanks to a survival biasing in which a transition is forced to occur before an exit of the safety domain or the end of the accident duration. When the SOMPEs are memorized, new families of unexpected events are defined, and independent batches of histories are played for each of them. Therefore, the total number of Monte Carlo runs increases (as the computation time does) in order to better assess dangerous circumstances consecutive to a failed protection.

In both cases however, there is a risk of wasting time in investigating accidental sequences whose probability is below a reasonable value of interest. Only negligible contributions to the score could result from costly CTB memorization phases or unexpected scenarios out of the SOMPEs. It can therefore be worth defining a probability threshold below which the corresponding family of dangerous evolutions is not further examined. A coherent approximation asks for the same value of this threshold whatever situation induces the potentially dangerous histories. In spite of this, in the case of a failed control device along an MPE, the memorization of the corresponding SOMPE is done, since it gives partial scores and possible paths to system failure in one try.

The coupling of MPEs and CTB memorization, and the SOMPEs have been implemented in a code dynamically estimating the unreliability of a complex system. Many tests have been run to show the performances of both approaches, and a parametric study of some results was done. All these results are presented and discussed in the next section.
4. Application

We have taken again the application of a PWR pressurizer, that was previously developed to illustrate the capabilities of the MPE-based algorithms [8]. For this problem, a two-region non-equilibrium thermal-hydraulic model is considered: the lower zone contains monophasic liquid water or saturated liquid water with bubbles, while the upper region consists in monophasic vapor or saturated vapor with droplets. The dynamic behavior of the pressurizer is described by 12 process variables. The main evolution equations are the mass and enthalpy balances in both zones, the remaining equations embodying the time dependent behavior of the control devices (spray system, proportional and back-up heaters, relief and safety valves), the heat losses through the walls and the evolution of the mean primary temperature. The dynamics of the pressurizer obviously depends on the state of each control means. An analysis of their failure modes was therefore carried on, revealing 900 different configurations for the whole system.

The initial values of the process variables correspond to the steady-state working conditions. Four possible initial states, in which this situation can be met, are considered. The initiating event provoking the transients under study consists in a quick diminution of the primary circuit mean temperature, followed by its return to its setpoint. The values of the parameters defining these variations have been chosen more realistic than in previous works. The system failure is defined by values of the pressure and pressurizer level triggering the reactor scram, or by its presence in unacceptable states, for which two control devices acting in the same way to mitigate pressure transients are simultaneously unavailable. At the end of the accident duration, the relief valve is asked to open three times, to finally mitigate the transient. Because of the relatively high value of the failure rate of this component, a merely piecewise-constant shape of the unreliability curve should be expected, thereby allowing us to check the estimations. But for smaller times, less probable accidental sequences can contribute to the final outcome, and a proper estimation of the associated risks is a test for the accuracy of the algorithms.

In order to give as a complete comparison as possible, the following simulation techniques have been applied to this problem:

- An analog simulation with efficient estimators scoring the probability of a system failure at each stage of a history rather than only when a failure is actually sampled (game ana).
- An analog game preceded by the memorization of the MPEs main characteristics, which are used to bypass the calculation of the most frequently followed trajectories, by sampling the stage of an MPE where an unexpected event takes place. This algorithm (game memo) is expected to be very fast, since most histories reach the end of the accident duration along an MPE, without requiring supplementary calculations. But there is no improvement in the investigation of rare events.
- On the basis of the previous techniques, an unexpected event is forced to occur in each history before the end of the accident duration, and the bias in the results is corrected by using the memorized information. This game (bias) requires new dynamic calculations in each history, and is therefore not very fast, but produces significantly more accurate results and allows the detection of less probable risks.
- The next algorithm consists in decomposing the simulation in independent estimations which are associated with each kind of unexpected events that may occur along each MPE and which are weighted by their respective probability. All not immediately mitigated transients are therefore taken into account (game sum).
• The same kind of game is played, but when the first unexpected event to occur is the failure of a control means, the corresponding SOMPE is memorized and independent estimations are performed on this basis, in a similar way to what is done for the first-order MPEs (game SOMPE).

• A CTB memorization is implemented in the two previous algorithms, for the simulation of scenarios following a first stochastic transition (games sum + CTB and SOMPE + CTB).

• Finally, a probability threshold $\epsilon$ is introduced in the game SOMPE + CTB, in order to save computer resources used to simulate histories associated with negligible risks (game lim-n for $\epsilon = 10^{-n}$).

In all games for which batches of histories are played independently for each type of accidental scenarios (games sum, SOMPE, ...), the number of histories in the $k$th batch while considering unexpected sequences out of the MPE started in state $i$ is [7]:

$$N_k^i = \int \left( \frac{\alpha + (1 - \alpha)n_{ue}^i p_k^i}{\sum_{l=1}^{n_{ue}^i} p_l^i} \right) + 0.5 \right), \quad 0 \leq \alpha \leq 1 \quad (3)$$

where $n_{ue}^i$ is the number of different possible unexpected events of a given kind out of the MPE started in state $i$ ($n_{ue}^i = k_{max}^i$ in case of a stochastic transition; $n_{ue}^i = k_{max}^i - 1$ in case of a failed control device), $p_k^i$ is the probability of the $k$th unexpected event, $\alpha$ is a real parameter ($\alpha = 0.2$ in the following) and $N_{min}^i$ is the reference number of histories for the batches corresponding to this MPE:

$$N_{min}^i = \begin{cases} N_{min} + \text{int}(\pi(i, 0) \cdot (N_{max} - N_{min}) + 0.5) & \text{in case of a stochastic failure}, \\ N_c & \text{in case of a failed control means}. \end{cases} \quad (4)$$

$N_{min}, N_{max}$ and $N_c$ are user-defined values, while $\pi(i, 0)$ is the initial probability of state $i$.

Table 1 gathers the number of histories and computer times for the first series of tests, in which the CTB and SOMPE parameters were: $N_{min} = 20$, $N_{max} = 100$, $N_c = 25$, $N_{nod}^{max} = 25$ (except for the game lim-n for which $N_c = 15$ and $N_{nod}^{max} = 10$), $N_{nod} = 6$ and $N_{nod}^{min} = 4$. The corresponding estimations of unreliability and their respective standard deviations are displayed in Figs. 1–6. As expected, the game memo is very fast, but almost blind to little probable risks, which are only correctly detected when independent batches of X histories are played. The allowance given to the SOMPEs increases both the computer time and number of histories, but for this application it does not seem to bring any significant

<table>
<thead>
<tr>
<th>Game</th>
<th>Number of histories</th>
<th>Computer time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ana</td>
<td>10000</td>
<td>3 h, 57 min, 11 s</td>
</tr>
<tr>
<td>memo</td>
<td>10000</td>
<td>16.4 s</td>
</tr>
<tr>
<td>bias</td>
<td>10000</td>
<td>1 h, 52 min, 42 s</td>
</tr>
<tr>
<td>sum</td>
<td>4010</td>
<td>28 min, 17 s</td>
</tr>
<tr>
<td>SOMPE</td>
<td>10999</td>
<td>41 min, 22 s</td>
</tr>
<tr>
<td>SOMPE + CTB</td>
<td>3973</td>
<td>24 min, 35 s</td>
</tr>
<tr>
<td>lim-6</td>
<td>3676</td>
<td>20 min, 08 s</td>
</tr>
<tr>
<td>lim-7</td>
<td>3973</td>
<td>20 min, 52 s</td>
</tr>
</tbody>
</table>
Fig. 1. Comparison of the estimations of unreliability.

Fig. 2. Comparison of the estimations of unreliability.

Fig. 3. Comparison of the estimations of unreliability.
Fig. 4. Comparison of the standard deviations on the estimations of unreliability.

Fig. 5. Comparison of the standard deviations on the estimations of unreliability.

Fig. 6. Comparison of the standard deviations on the estimations of unreliability.
change to the results. The CTB memorization leads to an interesting reduction in the required computer resources, for a given number of histories, while the introduction of a probability threshold turns out to be very efficient in suppressing costly meaningless histories. It can be observed that there is a noticeable difference between the results obtained in the CTB-based games and in their continuous correspondents (i.e. without CTB, see curves test 1 in Fig. 3). The reason for this gap is two-fold: first, an approximation is introduced through the interpolation done on the results of the CTB memorization; then, the survival biasing performed in the state the system evolves in after a first stochastic transition is likely to bring along more contributions to the score. The figures of merit of all games are presented in Figs. 7 and 8 and confirm our first conclusions.

A second series of test was performed, in which the same number of nodes in all sections of the MPEs was taken, whatever amount of time is elapsed during them. Table 2 shows that this entails an increase of computer time due to supplementary CTB-like memorization phases to be done for a given number of histories, but without really modifying the accuracy of the results.

Fig. 7. Comparison of the figures of merit.

Fig. 8. Comparison of the figures of merit.
Many runs of the different versions of the code were done to provide a sensitivity analysis of the results to some of the parameters in these methods.

The first study concerns the reference numbers of histories, and Table 3 gives the characteristics of the simulations run with $N_{\text{min}}=50$, $N_{\text{max}}=300$ and $N_c=40$. We see that the games with memorization of the SOMPEs strongly increase the number of histories, but for a relatively limited computer cost. A detailed study of the results (see curves test 3 in Fig. 3) shows that the gap in the estimations with and without CTB memorization tends to vanish, as much through a slight decrease of the results obtained with the CTB-based games as through a slight augmentation of the estimation in the $\text{sum}$ and $\text{SOMPE}$ games. Our previous conclusions on this subject could thus be confirmed.

Varying the value of the probability threshold is another sensitivity study worth to be performed. Table 4 displays the results of the simulations, performed only for the scenarios following either a stochastic failure or the failure of a control device. In this application, the estimations are not significantly modified, and relatively large values of $\epsilon$ can be taken into account to further reduce the computer time.

We have also considered the influence of the parameters used in the definition of the grid, as shown in Table 5 which concerns only the accidental sequences following a stochastic transition. Different values of $N_{\text{nod}}^{\text{min}}$, $N_{\text{nod}}^{\text{max}}$ and $\overline{N}_{\text{nod}}$ were taken, for an unchanged number of 2759 histories. Negligible modifications of the estimations were observed, thereby indicating that the values chosen for the first series of tests were too conservative. It also justifies the smaller value of $\overline{N}_{\text{nod}}$ that was entered in the $\text{lim-n}$ games.

Finally, we have investigated the influence of the $N_c$ parameter in the simulation of scenarios due to a failed control means (see Table 6). The allowance given to the SOMPEs allows to detect new sequences.
leading to the system failure, and that were not seen without this supplementary memorization step. Though the probability of these scenarios is very low in this specific application (reason why we do not present figures displaying this fact), it proves that a better exploration of the system possible evolutions after the failure of a control device is worth being realized.

5. Conclusions

This paper has presented a way to combine the CTB memorization and the MPE-based algorithms, in order to further improve the Monte Carlo simulation of a system unreliability in the frame of dynamic
reliability. This hybrid technique consists in defining on each section of the MPEs a grid of nodes, from which important characteristics of the system evolution in all possible new states are computed and kept in memory. Through an interpolation scheme on these memorized quantities, an acceleration of the simulation can be obtained and survival biasing can be played to better examine rare scenarios. For the same reason, the SOMPEs can be memorized after the failure of a control means. This operation increases the number of histories and computer time, but allows the detection of new risky situations. Meaningless failures (in terms of their probability of occurrence) can then be given up by introducing a probability threshold.

A large number of tests were performed on the application of a PWR pressurizer. They confirmed the expected advantages of the hybrid technique and of the SOMPEs, and provided a simple sensitivity analysis on most parameters of the simulation.

References