Procedures of Monte Carlo transport simulation for applications in system engineering

P.E. Labeau\textsuperscript{a,\ast}, E. Zio\textsuperscript{b}

\textsuperscript{a}Service de Métrologie Nucléaire, Université Libre de Bruxelles (CP165/84), 50, Avenue F.D. Roosevelt, B-1050 Brussels, Belgium
\textsuperscript{b}Department of Nuclear Engineering, Politecnico di Milano-34/3, Via Ponzio I-20133 Milan, Italy

Abstract

Monte Carlo (MC) simulation is the most promising tool for performing realistic reliability and availability analysis of complex systems. Yet, the efficient use of MC simulation technique is not trivial in large scale applications.

This paper considers the two commonly adopted approaches to MC simulation: the direct, component-based approach and the indirect, system-based approach. The mathematical details of the two approaches are worked out in detail, so as to show their probabilistic equivalence. The proper formulation for biasing the simulation is introduced, thus leading to the correct expressions for the statistical weights.

Both approaches are applied, in an analog as well as in a biased scheme, to a simple system of the literature and comparisons are made with respect to the computing time and the goodness of the estimate, as measured by the variance of the results. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Monte Carlo simulation; Biasing techniques; Transition time and transition probability

1. Introduction

The ongoing development of computer power has allowed a strong increase in the use of Monte Carlo (MC) methods for system engineering calculations. In the past, restrictive modeling assumptions had to be introduced to fit the models to the numerical methods available for their solution, at the cost of drifting away from the actual system operation and at the risk of obtaining sometimes dangerously misleading results. Thanks to the inherent flexibility of MC simulation, these assumptions can be relaxed, so that realistic operating rules, including, for example, maintenance policies and component aging processes, can be accounted for in the model. This is of fundamental importance for systems and plants, such as those in the nuclear industry, which are safety-critical and must be designed and managed within a risk-informed approach.

Various works [1–7] have highlighted the transport framework of failure-repair processes in system engineering. This provides a comprehensive formulation for the definition of indirect, system-based (SB) MC simulation algorithms: the next transition time of the whole system is firstly sampled, followed by the
has to be resampled after this transition. This can increase the numerical burden, and thus reduce the performances, of the CB approach. Moreover, the use of biasing techniques is not trivial in this case, because the effects of introducing modified probability laws for one component have to be accounted for at the whole system level when setting up the weights of the simulation.

This paper reviews both CB and SB approaches to MC simulation in a unified way, first in the case of independent components undergoing transitions to absorbing states only, and then when these assumptions are relaxed. A fundamental result of this unified presentation is the approach to the correct calculation of the statistical weights for both sampling procedures. The workload induced by the two approaches is discussed and compared, also when classical biasing techniques are applied.

The following structure of the presentation is here adopted: the unified mathematical basis of both MC simulation schemes is explained at length in Section 2; the performance of both approaches in analog games are compared in Section 3; in Section 4 the effect of biasing is considered on both algorithms. Some concluding remarks and perspectives are provided in Section 5.

2. Direct vs. indirect Monte Carlo: mathematical formulation

Let us consider a multi-component system made up of \( n \) independent units, each one having several different states. We begin by assuming, for the sake of simplicity and clarity, that, for each component, all states but the initial operating one are absorbing. This hypothesis allows us to account only for the first transition of each component, no matter which final state this component reaches. We denote by \( F_j(t) = 1 - S_j(t) \) the cdf of component \( j \) leaving its operational state before time \( t \), by \( f_j(t) \) the corresponding pdf, and by \( \bar{F}_j(t), \bar{S}_j(t) \) and \( \bar{f}_j(t) \) the corresponding biased laws [8,9].

2.1. Direct component-based Monte Carlo

Let \( t_j \) be the transition time sampled from \( f_j(t) \) for the \( j \)th component, \( j = 1, \ldots, n \). Reordering these sampled epochs according to their order of occurrence, we build the random walk of the system in this sampled history as

\[
(i_1, \tau_1), (i_2, \tau_2), \ldots, (i_n, \tau_n)
\]

where \( i_j \) is the \( j \)th component to undergo a transition and \( \tau_1 < \tau_2 < \ldots < \tau_k < T_m < \tau_{k+1} < \ldots < \tau_n \)

if we assume that \( k \) transitions are sampled before the mission time \( T_m \).

Let \( p((i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m) \) be the unbiased \( k \)-dimensional pdf of obtaining the sequence of events defined by Eqs. (1) and (2). It is obviously equal to:

\[
\begin{align*}
\mathbb{P}(i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m & = \prod_{j=1}^{k} f_j(\tau_j) \prod_{l=k+1}^{n} S_l(T_m) \\
\end{align*}
\]

(3)

If biased probability laws were used to sample the sequence of events (1) and (2), the biased pdf

\[
p((i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m)
\]

of obtaining such sequence of events would be formally identical to its unbiased counterpart (Eq. (3)), with \( f_j, S_j \) replaced by \( \bar{f}_j, \bar{S}_j \). Therefore, if \( w_0 \) is the initial value of the statistical weight associated to this MC run, its expression at \( T_m \) writes:

\[
w = w_0 \prod_{j=1}^{k} \frac{f_j(\tau_j)}{\bar{f}_j(\tau_j)} \prod_{l=k+1}^{n} \frac{S_l(T_m)}{\bar{S}_l(T_m)}
\]

(4)

A sequential interpretation of the sampled history (Eqs. (1) and (2)) can be given, using conditional probability laws to factorize Eq. (3) as:

\[
p((i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m)
\]

\[
= P((i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m | (i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > \tau_k)
\]

\[
\times P((i_1, \tau_1)\ldots(i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > \tau_k | (i_1, \tau_1)\ldots(i_{k-1}, \tau_{k-1}); \tau_{k-1}, \ldots, \tau_n > \tau_{k-1})
\]

\[
\times \ldots
\]

\[
\times P((i_1, \tau_1); (i_2, \tau_2); \tau_3, \ldots, \tau_n > \tau_2 | (i_1, \tau_1); \tau_2, \ldots, \tau_n > \tau_1)
\]

\[
\times P((i_1, \tau_1); (i_2, \tau_2); \ldots, \tau_n > \tau_1)
\]

(5)

Under this formulation, the random walk (Eqs. (1) and (2)) can be interpreted as a step-by-step sampling of the couples \((i_j, \tau_j)\). This will turn out to be useful for the comparison with the SB procedure (see the following), and when dependences between components are considered (see Section 2.3).

The expression of the various factors appearing in Eq. (5) can easily be constructed from the probability laws describing the behavior of the individual components [8]. Indeed:

\[
p((i_1, \tau_1); (i_2, \tau_2); \ldots, \tau_n > \tau_1) = f_i(\tau_1) \prod_{j=2}^{n} S_j(\tau_1)
\]

(6)
Note that Eq. (4) shows that the statistical weight associated with sampling from 1 when the assumptions of absorbing arrival states are released. When the weight is updated as the weight is updated as the probability, and not to a pdf or to a probability per unit time. By a capital letter, since it actually corresponds to a probability, and not to a pdf or to a probability per unit time. In case of biasing, the final statistical weight associated with the sampled history can also be factorized according to:

\[
w = w_0 w(i_1, \tau_1) \prod_{j=2}^{k} w(i_j, \tau_j) w_{l}(i_{k+1}, \ldots, i_{n})
\]

where

\[
f_{j}(	au_j) = \frac{n \prod_{j=2}^{n} S_{j}(\tau_j)}{n \prod_{j=2}^{n} S_{j}(\tau_j)}
\]

\[
w(i_1, \tau_1) = \frac{n \prod_{j=2}^{n} S_{j}(\tau_j)}{n \prod_{j=2}^{n} S_{j}(\tau_j)}
\]

\[
w(i_j, \tau_j) = \frac{n \prod_{j=2}^{n} S_{j}(\tau_j)}{n \prod_{j=2}^{n} S_{j}(\tau_j)}
\]

\[
l = 2, \ldots, k
\]

\[
w_{l}(i_{k+1}, \ldots, i_{n}) = \frac{n \prod_{j=k+1}^{n} S_{j}(T_{m})}{n \prod_{j=k+1}^{n} S_{j}(T_{m})}
\]

\[
\prod_{j=2}^{n} S_{j}(\tau_{j-1})
\]

\[
\prod_{j=2}^{n} S_{j}(\tau_{j-1})
\]

\[
\prod_{j=2}^{n} S_{j}(\tau_{j-1})
\]

\[
\prod_{j=2}^{n} S_{j}(\tau_{j-1})
\]

Note that Eq. (4) shows that the statistical weight associated with a whole history up to \(T_m\) is the product of the individual weights of the (independent) components’ transition events, whereas the factorization (Eqs. (9)–(12)) above gives rise to a more collective effect when the weight is updated as the history develops. This latter approach is easier to implement when the assumptions of absorbing arrival states are released.

Finally, the simulation workload is limited to the sampling from \(n\) one-dimensional time distributions and to a ranking procedure of the sampled times, in the unbiased case, whereas additional workload is brought in by the calculation of the weights, in the biased case.

### 2.2. The indirect system-based algorithm

Consider again the sampled sequence of events (1) and (2). Instead of building it by ordering the sampled transition times for each component, we can instead think of sampling first the time \(\tau_1\) at which the system will undergo its first transition, then, conditionally on \(\tau_1\), the component \(i_1\) and its arrival state, which determine the system configuration change, and so on up to the \(k\)th transition occurring before \(T_m\) [6]. This sampling process mathematically corresponds to a different decomposition of the pdf (5), that is now written as:

\[
p((i_1, \tau_1) \ldots (i_k, \tau_k); \tau_{k+1}, \ldots, \tau_n > T_m)
\]

\[
(f_1(\tau_1) q(i_1 | \tau_1) f_2(\tau_2 | (i_1, \tau_1)) q(i_2 | (i_1, \tau_1), \tau_2) \ldots \times (1 - F_s(T_m | (i_1, \tau_1) \ldots (i_k, \tau_k)))
\]

In the latter expression, \(f_s(\tau_1)\) is the probability density of the system having a transition at \(\tau_1\), independently of which transition actually occurs. Therefore, from Eq. (6) we find:

\[
f_s(\tau_1) = \sum_{l=1}^{n} f_s(\tau_l) \prod_{j=2}^{n} S_{j}(\tau_j)
\]

The corresponding cdf is obviously:

\[
F_s(\tau_1) = 1 - \prod_{l=1}^{n} S_{l}(\tau_l)
\]

since

\[
f_s(\tau) = - \frac{dS_s(\tau)}{d\tau}
\]

The probability that the transition occurring at \(\tau_1\) is that of component \(i_1\) is:

\[
q(i_1 | \tau_1) = \frac{\lambda_{i_1}(\tau_1)}{\sum_{j=1}^{n} \lambda_{j}(\tau_1)}
\]

where

\[
\lambda_{j}(\tau) = \frac{f_{j}(\tau)}{S_{j}(\tau)}
\]

is the value of the failure rate of component \(i_j\) at time \(\tau\). Introducing Eq. (17) in Eq. (16), we obtain after some transformations:

\[
q(i_1 | \tau_1) = \frac{f_s(\tau_1) \prod_{j=2}^{n} S_{j}(\tau_1)}{\sum_{j=1}^{n} f_s(\tau_j) \prod_{l=j}^{n} S_{l}(\tau_l)}
\]
From Eqs. (6), (14) and (18), it is easy to verify that the direct and indirect sampling procedures for \((i_1, \tau_1)\) are equivalent in probability, i.e.

\[
f_s(\tau_1)q(i_1|\tau_1) = p(i_1, \tau_1); \tau_2, \ldots, \tau_n > \tau_1
\]

For the successive transitions \(l = 2, \ldots, k\) of the random walk, we find:

\[
f_s(\tau_l|i_1, \tau_1) = \frac{\sum_{j=1}^{n} f_l(\tau_l) \prod_{m \neq j} S_m(\tau_l)}{n \prod_{m \neq j} S_m(\tau_{l-1})} \quad \text{for} \quad \tau_1 < \tau_l < \tau_{l-1}
\]

\[
f_s(\tau_l|i_1, \tau_1) = \begin{cases} 0 & \text{if} \quad \tau_l = \tau_{l-1} \\ 1 - \frac{1}{n} \sum_{j=1}^{n} S_j(\tau) \frac{S_j(\tau_{l-1})}{S_j(\tau_{l-1})} & \text{if} \quad \tau_l > \tau_{l-1} \end{cases}
\]

\[
q(i_l|i_1, \tau_1) = \frac{\lambda_l(\tau)}{n \sum_{j=1}^{n} \lambda_j(\tau_l)}
\]

\[
f_s(\tau_l|i_1, \tau_1) = \frac{\sum_{j=1}^{n} f_l(\tau_l) \prod_{m \neq j} S_m(\tau_l)}{\sum_{j=1}^{n} f_l(\tau_l) \prod_{m \neq j} S_m(\tau_l)}
\]

(22)

From Eqs. (7), (20) and (22), we can again verify the step-by-step probabilistic equivalence of the direct and indirect sampling procedures:

\[
f_s(\tau_l|i_1, \tau_1)q(i_l|i_1, \tau_1) = p(i_l, \tau_l); \tau_{l+1}, \ldots, \tau_n > \tau_l
\]

\[
(i_l, \tau_l)\ldots(i_{l-1}, \tau_{l-1}); \tau_l, \ldots, \tau_n > \tau_{l-1}
\]

(23)

Finally, from Eq. (21), we see that:

\[
1 - F_s(T_m|i_1, \tau_1)\ldots(i_k, \tau_k) = \frac{n}{\sum_{j=k+1}^{n} S_j(T_m)}
\]

(24)

in accordance with Eq. (8).

In case of biasing, the expressions (14), (18), (20)–(22), and the corresponding biased ones with \(f^\prime_l\) and \(S^\prime_l\), can be used during the random walk to update the weight of the current history at each sampling of a transition time \(\tau_l\) and corresponding component \(i_l\).

Though both direct, CB, and indirect, SB, approaches to constructing the random path (Eqs. (1) and (2)) of the system in phase space have been shown to be probabilistically equivalent, and simply resulting from two distinct decompositions (i.e. Eqs. (5) and (13)) of the same multivariate pdf, they are likely to differ in the numerical workload they entail. For a given sequence of \(k\) transitions occurring before \(T_m\), the SB approach requires \(2k + 1\) samplings: \((i_j, \tau_j)|\tau_j < T_m, j = 1, \ldots, k\) and \(\tau_{k+1} > T_m\). For the direct, CB, simulation, \(n\) samplings are required, no matter what the length of the random path up to \(T_m\) is (the number of samplings increases when non-absorbing component states are considered (see Section 2.4)). If a small number of components are likely to change their state before the mission time, the direct sampling procedure does not appear advantageous. Yet, this first conclusion might no longer apply for a biased game devised to increase the number of transitions occurring before \(T_m\). A fair comparison should then be made between the number \(n\) of component samplings in the CB procedure and the minimum number of transition samplings in the SB approach, that are compulsory in a ‘good’ biased game to attain the desired significance in the results. Moreover, one should also take into account the numerical complexity of sampling from system transition time distributions like Eq. (15) or (21), as compared with the numerical workload necessary for the sampling of the cdfs of individual components.

2.3. Accounting for dependences

Up till now, we have considered systems made of independent components. As an important consequence of this assumption, the probability laws of all components are unaltered by the outcomes of previously occurring transitions. In real systems, however, the hypothesis of independence rarely holds. For instance, in a system made of two components in active redundancy, the failure of one of them is likely to provoke a supplementary load on the second, whose proneness to fail may tend to increase. In the same way, such a trend would be observed after a first failure in a two-component system in warm standby.

Whatever type of dependence is introduced in the modeling of the system, any transition undergone by a single component may alter the probability laws of other units. In principle, the SB MC procedure does not suffer any limitation from the introduction of such dependences. Indeed, the times and types of transitions are sampled in succession from the current system configuration which accounts for the possible alterations in the components’ behavior. On the contrary, the direct, CB sampling procedure needs to be modified: the set of transition times initially sampled for all the components do not remain valid throughout the mission time, because the system random walk is altered during its evolution by the dependences. In other words, the sequence of components transitions sampled may need modifications because the occurrence of some of the transitions in the sequence modifies the probability of the successive transitions, thus requiring resampling.

Let then \(f^{(h)}(t)\) be the pdf of component \(j\) having a transition at the time \(t\), when the current system configuration is the result of the first \(l\) transitions along the random
path (Eqs. (1) and (2)). We thus have \( f_{i}^{(0)}(t) = f_{j}(t) \forall j \) from which the first set of transition times is sampled. Let \( \tau_{1} \) be the minimum of the sampled times, and \( i_{1} \) the component correspondingy changing its state at that time. Then, the proper continuation for an unbiased system history in the direct, CB approach would imply discarding the other samples and proceeding to the sampling of the next transition time of all components but \( i_{1} \) (in the hypothesis of all absorbing states except the initial, operating one), provided the sampled epochs are larger than \( \tau_{1} \). This amounts to sampling from the following truncated pdfs:

\[
\tilde{f}_{j}^{(1)}(t) = \begin{cases} 
0 & \text{if } t \leq \tau_{1} \\
\frac{f_{j}^{(1)}(t)}{S_{j}^{(1)}(\tau_{1})} & \text{if } t > \tau_{1} \ , \ j \neq i_{1}
\end{cases}
\] (25)

From this second sampling, the minimum transition time \( \tau_{2} \), corresponding to a transition of component \( i_{2} \), is conserved. In the same way, the \( l \)th sampling of transition times will be performed with the pdfs:

\[
\tilde{f}_{j}^{(l-1)}(t) = \begin{cases} 
0 & \text{if } t \leq \tau_{l-1} \\
\frac{f_{j}^{(l-1)}(t)}{S_{j}^{(l-1)}(\tau_{l-1})} & \text{if } t > \tau_{l-1} \ , \ j \neq i_{1}, \ldots, i_{l-1}
\end{cases}
\] (26)

In this case, only the sequential interpretation (5) of the direct, CB, approach holds. The truncated pdfs (26), and the corresponding values of \( S_{j}^{(l-1)}(\tau_{l-1}) \), \( t > \tau_{l-1} \), must then be used in the conditional probability laws (7) and (8). Since

\[
\tilde{S}_{j}^{(l-1)}(\tau_{l-1}) = \frac{S_{j}^{(l-1)}(\tau_{l-1})}{S_{j}^{(l-1)}(\tau_{l-1})} \quad \text{if } t > \tau_{l-1} \ , \ j \neq i_{1}, \ldots, i_{l-1}
\]

we easily find that

\[
p((i_{1}, \tau_{1}) \ldots (i_{l}, \tau_{l}); \tau_{l+1}, \ldots, \tau_{n} > \tau_{l})
\]

\[
(i_{1}, \tau_{1}) \ldots (i_{l-1}, \tau_{l-1}); \tau_{l+1}, \ldots, \tau_{n} > \tau_{l-1})
\]

\[
\tilde{f}_{i}^{(l-1)}(\tau_{l}) \prod_{j=1}^{n} \tilde{S}_{j}^{(l-1)}(\tau_{l}) = \frac{f_{i}^{(l-1)}(\tau_{l}) \prod_{j=1}^{n} S_{j}^{(l-1)}(\tau_{l})}{\prod_{j=1}^{n} S_{j}^{(l-1)}(\tau_{l-1})} \ ,
\]

\( l = 2, \ldots, k \) (28)

The expressions (10)–(12) of the statistical weights are thus still valid if, at each stage of the sampling of the random walk, the current forms of the pdfs of the components are used.

This adaptation of the sampling procedure obviously works, but it entails a number of additional samplings which could quickly turn out to be costly if many components are affected by previous transitions. One could, thereby, think of keeping the first series of sampled transition times, and of using them directly in a game that is inevitably biased. The actual pdf of a transition of component \( j \), before and after component \( i_{1} \) has undergone a transition at \( \tau_{1} \), is:

\[
f_{j}^{(0)}(t) = \begin{cases} 
0 & \text{if } t \leq \tau_{1} \\
\frac{S_{j}^{(0)}(\tau_{1})}{S_{j}^{(1)}(\tau_{1})} f_{j}^{(1)}(t) & \text{if } t > \tau_{1} \ , \ j \neq i_{1}
\end{cases}
\] (29)

Indeed, component \( j \) will undergo a transition after \( \tau_{1} \), if it has kept evolving in its initial state up to \( \tau_{1} \), at which time its transition probability changes as given by Eq. (25) due to the occurrence of the transition of component \( i_{1} \). Then, by generalizing Eq. (30), we find for the actual pdf of component \( j \) having a transition after \( l \) previous transitions:

\[
f_{j}^{(l)}(t) = \begin{cases} 
0 & \text{if } t \leq \tau_{1} \\
\frac{S_{j}^{(l)}(\tau_{l})}{S_{j}^{(l+1)}(\tau_{l})} f_{j}^{(l+1)}(t) & \text{if } \tau_{l} < t \leq \tau_{l+1} \\
\vdots & \text{if } \tau_{l} < t \leq \tau_{l+1} \\
\frac{S_{j}^{(l)}(\tau_{l})}{S_{j}^{(l+1)}(\tau_{l})} f_{j}^{(l)}(t) & \text{if } t > \tau_{l}
\end{cases}
\] (31)

We can, therefore, think of a biased simulation in which we use the first set of transition times sampled from the \( f_{j}^{(0)}(t) \), and compute statistical weights according to Eqs. (10)–(12), provided we replace the actual pdfs \( f_{j}(t) \) by \( f_{j}^{(l)}(t) \), and the biased pdfs \( \tilde{f}_{j}(t) \) by \( \tilde{f}_{j}^{(l)}(t) \).

However, it is interesting to note that in the case of the two-component redundant system mentioned above as an illustration of dependence, conserving the transition time of the second component, which was sampled from the initial pdf \( f_{2}^{(0)}(t) \), would amount to delaying the second failure—since the overload after the first component failure would increase the likelihood of failure of the second one—thereby,
biasing the simulation in the opposite direction to the system failure which is a rare event of interest. To perform an effective biasing scheme, then, the initial set of samplings should be realized with pdfs different from those corresponding to the nominal state of the components.

2.4. The general case with non-absorbing states

In this Section, we go back to the assumption of no dependences, but release that of all component states but the initial ones being absorbing. This entails several consequences in the CB approach. First, after a component \( j \) has undergone a transition, it still has to be considered in the sequel of the random walk history. A new time has to be sampled for the transition of this component from the new state \( \alpha_j \) it lies in as a result of its last occurred transition. The probabilistic laws governing the behavior of component \( j \), thereby, depend on its current state \( \alpha_j \). Moreover, the argument of these time distributions is no longer the ‘absolute’ time, but the time interval elapsed since component \( j \) underwent its last occurred transition. The probabilistic interpretation of the sampling, since this time the same component can change its state more than once before \( T_m \) and appear several times in the random walk.

If some components have more than two states, the definition of the random walk must be complemented by giving, at the \( n \)th step, the state that component \( i \) has a transition to. However, this does not add any difficulty in the probabilistic interpretation of the sampling, since this type of event corresponds to the sampling of a discrete distribution with a limited number of states. For this reason, in the following, we will treat binary components, which avoid us to account for these additional simple samples.

In the direct, CB simulation algorithm, the pdf of sampling the random walk \( (i_1, \tau_1)(i_2, \tau_2), \ldots, (i_k, \tau_k), \tau_k < T_m < \tau_{k+1} \) now becomes (see Eq. (5)):

\[
p((i_1, \tau_1), \ldots, (i_k, \tau_k); \tau_{k+1}, \ldots, \tau_{k+n-1} > T_m) = P((i_1, \tau_1), \ldots, (i_k, \tau_k); \tau_{k+1}, \ldots, \tau_{k+n-1} > T_m) \times P((i_1, \tau_1), \ldots, (i_{k-1}, \tau_{k-1}); \tau_{k+1}, \ldots, \tau_{k+n-1} > \tau_{k-1}) \times \ldots \times P((i_1, \tau_1), (i_2, \tau_2), \tau_3, \ldots, \tau_{n+1} > \tau_2) \times ((i_1, \tau_1); \tau_2, \ldots, \tau_n > \tau_1)p((i_1, \tau_1); \tau_2, \ldots, \tau_n > \tau_1)
\]

with the following adaptation of expressions (6)–(8):

\[
p((i_1, \tau_1); \tau_2, \ldots, \tau_n > \tau_1) = f_{i_1}(\alpha_{i_1}, \tau_1 - \tau_1^0) \prod_{j=1}^n S(\alpha_j, \tau_1 - \tau_j^0)
\]

and

\[
p((i_1, \tau_1), \ldots, (i_k, \tau_k); \tau_{k+1}, \ldots, \tau_{n+k-1} > T_m) = \prod_{j=1}^k S(\alpha_j, T_m - \tau_j^0)
\]

Note that Eq. (35) has the same form as Eq. (24).

On the other hand, in the system-based approach, Eq. (13) is still valid. Yet Eqs. (14) and (16) have to be adapted in the following way:

\[
f_i(\tau_1) = \sum_{l=1}^n f_i(\alpha_l, \tau_1 - \tau_l^0) \prod_{j=1}^n S(\alpha_j, \tau_1 - \tau_j^0)
\]

and

\[
g(i_1|\tau_1) = \frac{\lambda_i(\alpha_{i_1}, \tau_1 - \tau_{i_1}^0)}{\sum_{j=1}^n \lambda_j(\alpha_j, \tau_1 - \tau_j^0)}
\]

In the same way, we have for Eq. (20):

\[
f_i((i_1, \tau_1), \ldots, (i_{k-1}, \tau_{k-1})) = \sum_{m=1}^M f_i(\alpha_m, \tau_1 - \tau_m^0) \prod_{j=1}^n S_m(\alpha_m, \tau_1 - \tau_m^0)
\]

Finally, Eq. (22) is similar to Eq. (37), if one replaces \( i_1, \tau_1 \) with \( i_2, \tau_2 \). The case of Section 2.3 can also be easily generalized to this case.

3. Numerical example

Consider a supply system (see Fig. 1) made of three identical pumps \( p \) in parallel, with a 2-out-of-3 logic, and a
valve (v) in series [10,11]. Failure densities are Weibull distributions
\[ f_i(t) = \beta_i \alpha_i \exp(-t/\alpha_i)^{\beta_i-1}, \quad i = p, v \]
while repair densities are Erlang distributions of order 2:
\[ g_i(t) = \rho_i t^{\rho_i-1} e^{-\rho_i t}, \quad i = p, v. \]
All components are as good as new at time 0.

If the components are independent, the system unavailability \( Q(t) \) at time \( t \) reads:
\[ Q(t) = 1 - A(t) = 1 - (1 - Q_i(t))(1 - Q_p(t))^2(1 + 2Q_v(t)) \]
where \( Q_i(t) \) is the unavailability of component \( i \), if it is as good as new at \( t = 0 \). Moreover, if \( \beta_i = 1 \), \( f_i(t) \) reduces to an exponential pdf of parameter \( \lambda_i = 1/\alpha_i \). The expression of \( Q(t) \) can then be analytically obtained, from the classical equations of an alternate renewal process [12]:
\[
1 - Q_i(t) = A_i(t) = 1 - \int_0^t f_i(\tau) d\tau + \int_0^t A_i(t-u) du \int_0^u f_i(\tau) g_i(u-\tau) d\tau
\]
Taking the Laplace transform of the latter expression, we find after some calculations:
\[
Q_i(t) = \begin{cases} 
Q_{i,w} \left( 1 + \left( \frac{3A_i}{2} - 1 \right) e^{-3A_i/2} \right) & \text{if } \rho_i = \frac{\lambda}{4} \\
Q_{i,w} \left( 1 + \left( \frac{\lambda - \rho_i}{2\delta_i} \sin \delta_i t - \cos \delta_i t \right) e^{-\left( \lambda/2 + \rho_i \right) t} \right) & \text{if } \rho_i > \frac{\lambda}{4} \\
Q_{i,w} \left( 1 + \left( \frac{\lambda - \rho_i}{2\delta_i} \sinh \delta_i t - \cosh \delta_i t \right) e^{-\left( \lambda/2 + \rho_i \right) t} \right) & \text{if } \rho_i < \frac{\lambda}{4} 
\end{cases}
\]
where
\[ Q_{i,w} = 2\lambda_i/(2\lambda_i + \rho_i), \quad \delta_i = \sqrt{[\lambda_i(\lambda_i/4 - \rho_i)]} \]
Assume all components have the same MTTF = \( \alpha_i \Gamma(1 + 1/\beta_i) = 100 \) h, and the same MTTR = 2/\( \rho_i = 100 \) h. For \( \beta_i = 1, \ i = p, v \), and a mission time of 400 h, Fig. 2 compares the analytical solution with the estimations of the system unavailability obtained by the CB and SB MC analog algorithms. The validation and equivalence of both algorithms is, thereby, clearly illustrated.

In a second series of tests, we have adopted a value of \( \beta_p = 1.5 \) and \( \beta_v = 1.2 \), with the same MTTF as above. The unreliability and unavailability results obtained with the two MC algorithms, not reported here for the sake of brevity, are equivalent. The computer times required for the two simulations with 10^5 histories are given in Table 1.

The system unavailability (unreliability) is estimated using the classical 0–1 estimator, i.e. scoring a contribution \( H(t - t_i) \) for a failure taking place at \( t_i \), and \( -H(t - t_i) \) if the system is repaired at \( t_i \) (for the unavailability estimation, of course), \( H(\cdot) \) being the Heaviside function. A contribution to the score can then only be obtained after the complete sampling of a pair of events \( (i, t_i) \). As both approaches are probabilistically equivalent, the variance of the score is the same for both games. Therefore, the efficiency of the MC games is measured simply by comparing the computer times needed. On this basis, the CB game appears to be more efficient. Since this approach strongly depends on the number of components in the system, Table 2 presents the computer times needed for the unavailability estimation of a system made of 7 pumps in parallel (5-out-of-7 logic), in series with a 2-out-of-3 block of valves, all the components having the same characteristics as before. Again for 10^5 histories, the SB approach is even less efficient in this case. This is due to the greater numerical intricacy of inverting the system failure density, as compared with the individual samplings of the Weibull failure pdfs of the components, thus making a large number of CBS samplings faster.

4. Biasing techniques

4.1. Exponential biasing with mean times preserved

To make all samplings straightforward, all failure and repair times can be sampled from exponential distributions, preserving the mean of the failure and repair densities [13]. By so doing, these biased distributions aim only at simplifying the sampling procedure, and not at forcing specific events to occur. Yet the computational effort required is simply shifted from sampling non-exponential distributions to computing statistical weights (based on expressions (6)–(16)). Our interest then focuses on whether this shift influences the relative performances of our two approaches or not. The considerations made in Section 3 on considering only the computer times to compare the two MC algorithms are still valid here. Yet the variances of their estimates should also be compared with those obtained with the analog games.

This exponential biasing has been tested on the four-component system of Section 3, with the same values of the parameters of the Weibull and Erlang distributions. Table 3 gathers the computer times, for 10^5 histories, required to estimate the system unavailability with the two different MC approaches.

As expected, the improvement is in the SB approach in which the costly samplings from the intricate system pdf are...
substituted by the trivial exponential samplings, whereas for the CB algorithm the samplings of the components failure times by direct inversion of the Weibull distribution were already straightforward in the corresponding analog game so that no gain in computing time is achieved. Actually, the small additional computer time is due to the calculation of the statistical weights.

Fig. 3 shows the variances obtained for both biased games, as well as for the analog simulations (see Section 3). The savings in sampling gained with the biasing to exponential distributions do significantly alter the accuracy of the simulations. Moreover, one can observe the instability of the estimations obtained: for mission times of the order of a few rise times of the unavailability curve, the estimation of the asymptotic limit is not correct and a variance increasing with time is obtained. This phenomenon, illustrated by the trend of the curves in Fig. 3, was not observed for analog games. Indeed, the unitary positive and negative contributions to the score obtained in the latter simulations compensate each other as soon as the asymptotic value $Q_1 = \frac{1}{2} A_1$ of the unavailability is reached. This is a classical result, the score obeying a Bernoulli distribution and the variance being then equal to $Q_1 (1 - Q_1)$. This is no longer true when unequal positive and negative contributions are scored, corresponding to the current value of the weight at the time the system fails and is repaired, respectively. Such a behavior of a linearly growing variance of estimations obtained with estimators other than the 0–1 case was previously shown [14].

4.2. Forcing rare events

Consider now the same four-component system as before, but with more reliable characteristics: an MTTF = $10^6$ h is chosen for the pumps (their MTTR remaining unchanged), while the valve has an MTTF = $10^7$ h and an MTTR = 24 h. In such a case, the analog simulations, or the biased games with the exponential biasing, lead to an underestimation of the system unavailability, for a limited number of MC histories. Indeed, the system reliability is such that too few system failure events are likely to be sampled within the mission time $T_m$ of a few hundreds of hours considered. A classical way to overcome this problem in Markovian reliability analysis consists in forcing the occurrence of the next event before $T_m$ [1], with a corresponding updating of the statistical weights along the history. If the system unavailability is estimated, this biasing technique prevents a history from finishing. It must then be combined with a Russian roulette, in order to suppress runs with an excessively small weight. Let us see now what this technique implies in both SB and CB approaches.

4.2.1. System-based approach

Forcing a transition before $T_m$ amounts to sampling the

---

Table 1

<table>
<thead>
<tr>
<th></th>
<th>Component-based</th>
<th>System-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unreliability (s)</td>
<td>1.2</td>
<td>9.3</td>
</tr>
<tr>
<td>Unavailability (s)</td>
<td>4.6</td>
<td>50.0</td>
</tr>
</tbody>
</table>

---

Fig. 2. Comparison of the two MC estimations with the analytical solution.
The statistical weight corresponding to this forced sampling is:

\[ w = F_s(T_m|\tau) = 1 - \frac{1}{n} \sum_{j=1}^{n} \frac{S_j(T_m - \tau^j)}{S_j(T - \tau_j)} \]  

(44)

which is equivalent to the classical result of Markovian reliability analysis.

4.2.2. Component-based approach

Performing this type of biasing in the CB approach could lead to sampling the next transition time of each component and rejecting the full sample if all transition times are larger than \( T_m \). Obviously, for a reliable system, this algorithm would be very poor in efficiency, since many samplings would be rejected. A better way of biasing the simulation consists in forcing each component to undergo its next transition before \( T_m \):

\[ \tilde{f}_s(t|\tau) = f_s(t|\tau)/F_s(T_m|\tau) \quad \text{if} \quad t \leq T_m \]  

(43)

The statistical weight corresponding to this forced sampling is:

\[ w = F_s(T_m|\tau) = 1 - \prod_{j=1}^{n} \frac{S_j(T_m - \tau^j)}{S_j(T - \tau_j)} \]  

(44)

shown that \( w \) can be written as:

\[ \prod_{j} \left( 1 - \frac{S_j(T_m - \tau^j)}{S_j(T - \tau_j)} \right) \]  

(45)

if \( t \) is the minimum transition time among all components corresponding to component \( i \).

Actually, this modified form of the classical biasing scheme given in Ref. [1] presents the following conceptual advantage: all components are likely to undergo a transition, no matter what their reliability characteristics are. This means that low probable events can be sampled more easily than in the SB approach, where an unbiased sampling of the component undergoing the transition at \( t \) will always favor repairs, at the expense of more interesting situations with multiple component failures. For this reason, we have also considered a double biasing of the SB algorithm, in which the sampling of the component having a transition is done from an equiprobable distribution.

Simulations with \( 10^6 \) trials have been performed with these three biased games, in the limit case where \( \beta_i = 1, i = p, v \), and for \( T_m = 200 \) h. Table 4 gathers the computer times obtained for each of them. Once again, the CB game turns out to be the fastest. Fig. 4 displays the different estimates of the system unavailability, while Fig. 5a and b compare the variances of the three biased games.

---

Table 2: Computer times for the 10-component system

<table>
<thead>
<tr>
<th>Component-based</th>
<th>System-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unavailability</td>
<td>11.2 s</td>
</tr>
</tbody>
</table>

Table 3: Computer times for the four-component system with exponential biasing

<table>
<thead>
<tr>
<th>Component-based</th>
<th>System-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unavailability</td>
<td>11.9 s</td>
</tr>
</tbody>
</table>

---

Fig. 3. Variances of the analog estimations and of the estimations with exponential biasing.
whole mission time and in the first 140 h, respectively. We observe that the SB estimation with double biasing turns out to be the closest to the exact solution. For most of the mission time, the CB algorithm leads to a significantly smaller variance than both SB games, but for times closer to $T_m$, its accuracy deteriorates significantly. This result is not surprising: as we keep sampling transition times before $T_m$ for all components, we obtain a large number of transitions for the whole system in the time interval just before $T_m$; this leads potentially to a large number of contributions to the score, but with very different values of the statistical weights. This situation might be improved by stopping the biasing as soon as the weight reaches a minimal threshold, or when the current time in a history gets close to the mission time. On the other hand, the SB approach with the additional biasing of the transition probabilities is obviously improved on the first part of the mission time. However, the variance in the SB game with no biasing of the transition probabilities is much more stable on the whole time interval. Indeed, unfavoring the probable reparations of the components corresponds to statistical weights larger than 1, which can degrade the variance of the estimates. This trend is likely to be reduced if smaller values of the biased transition probabilities leading to additional failures are taken.

5. Conclusions

Monte Carlo simulation is establishing itself more and more as the most promising approach to many system engineering applications, as a more integrated view of the different aspects of the problem is beginning to prevail among practitioners. Yet the high reliability of hazardous industrial systems demands for efficient biasing algorithms, which, in turn, ask for a deep understanding of the mathematics underlying the simulation.

Two approaches to MC algorithms are available for the simulation of the evolution of a system. Their overall performances have been compared in this paper, both when used in an analog form and when combined with classical biasing techniques.

More precisely, we have investigated the case where the component failure time pdfs are analytically invertible, making the CB sampling straightforward, while the SB may suffer the sampling from an intricate failure density. The CB game turns out to be more efficient in such circumstances, even for a large number of components or when the sampling is performed from biased exponential distributions preserving the MTTF (or MTTR) of the components. However, in the latter case two observations can be made: first, the computer times required by both CB and SB simulation approaches get closer to each other, the CB case being still the fastest one; secondly, the variance of these biased games is much worse than those of their respective analog correspondents.

When the system is made of reliable components, transitions must be forced before the mission time. This classical technique was directly adapted to the SB case, while a different approach was used in the CB algorithm.

Again, the latter turned out to be more efficient than the SB game, even when for the SB simulation the transitions forced before the mission time are coupled with a biased sampling of the component changing its state.

<table>
<thead>
<tr>
<th>Component-based</th>
<th>System-based</th>
<th>System-based (double biasing)</th>
</tr>
</thead>
<tbody>
<tr>
<td>47.6 s</td>
<td>181.8 s</td>
<td>144.8 s</td>
</tr>
</tbody>
</table>

Table 4

Computer times for the four-component system with forced transitions

Fig. 4. Estimations of the unavailability when transitions are forced before $T_m$. 
In the various tests performed, the CB games appeared to be more efficient, in spite of the fact that the statistical weights in biased games are always computed at the system level. Yet these primary results obtained on a test case should be confirmed and completed on large-scale applications.

References


