F.Bourgeois [*]& P. E. Labeau [†]
Université Libre de Bruxelles, Belgium

ABSTRACT: PSA results are expected to be used to improve the design, the operating procedures or the safety policy of a system. It then appears logical to combine a risk analysis with an optimisation procedure, aiming at minimising a cost function to determine the most appropriate strategy. This idea is applied in this work to dynamic systems, i.e. installations for which the probabilistic risk analysis must be conducted using dynamic reliability methods. As several system parameters can be uncertain, the cost function turns out to be a random variable. Stochastic programming techniques are convenient for such calculations. The whole optimisation procedure is presented, and a first application of the algorithm on a HNO_3 supply system is provided.

1 INTRODUCTION

Dynamic reliability (Siu 1994; Devooght 1997) is gaining more and more recognition in the reliability community as a more appropriate methodology to perform probabilistic safety studies of systems in which dynamic aspects of the accidental transients turn out to be essential. Due to the numerical complexity of the problem, Monte Carlo (MC) simulation stands out as one of the most convenient solution scheme for large installations, provided information from very probable faultless evolutions is used to drive the MC algorithm towards transients with potential damage (Labeau 1996; Labeau 1997).

As experience in performing PSA studies grows, using their results in decision making, system management or design optimisation becomes a main topic of interest. For severe accidental transients, this issue can be related with the choice of an optimal safety procedure. This includes e.g. a choice of equipment, or the definition of a threshold on a critical process variable, which corresponds to the triggering of a protection device. This last situation is of practical importance: a quick shutdown of the system is conservatively safe, but it entails more regularly a loss of revenue due to the interruption in the production; on the other hand, a delayed reaction to the development of a transient can significantly reduce the safety margin of the system, and hence affect the probability of an important cost associated with a damage. Having this economical context in mind, the present work deals with pioneer efforts in order to couple optimisation techniques with a dynamic approach to reliability.

The problem under study amounts to minimising a cost function, the definition of which includes both the investment cost to implement a given safety policy (refered to as a strategy in the following), and the cost induced by a damage in case of system failure or by the loss of revenue following a system shutdown. This cost function depends on the safety characteristics of the system. Whatever strategy is considered, uncertainties on system parameters make the cost function random. One could minimise the expected value of this function, hence obtaining again a classical optimisation problem. But in actuality, the designer is more interested in being as sure as possible that the total cost will not exceed a given limit. The optimal strategy is thus that maximising the probability of such event.

This paper is organised as follows. Section 2 is dedicated to the mathematical expression of the optimisation problem, and to the numerical aspects of the algorithm used to solve it. The whole procedure is applied in section 3 to a HNO_3 supply system, which displays a dynamic behaviour for the kind of transients that we will consider. Results from several numerical tests are presented and commented. Some conclusions and perspectives are briefly discussed at the end of the paper.

[*] Research Assistant, National Fund for Scientific Research (Belgium). On leave at Stanford University, Department of Mathematics

[†] Senior Research Assistant, National Fund for Scientific Research (Belgium)

2.1 Mathematical definition of the problem

Consider a vector $\bar{u}$ of strategies, which are actually parameters determining the safety policy adopted for the system. These strategies belong to a given space $U \subset R^m$. Our optimisation problem consists in minimising on all values of $\bar{u}$ a cost function ϕ. If we plan to select the most appropriate safety policy to mitigate a dynamic transient in the system, we must account in the definition of ϕ for different kinds of costs:

- the investment budget ϕ_{inv} necessary to implement the chosen strategy;
- the cost ϕ_s induced by a system shutdown, and
- the damage ϕ_f entailed by a system failure.

Therefore, the total cost is expressed as

$$\phi = \phi_{inv} + p_s \phi_s + p_f \phi_f \qquad (1)$$

where p_s and p_f are the probability of a system shutdown and that of its failure, respectively. All these quantities depend on the strategy. But for a given $\bar{u}$, ϕ turns out to be a random function, because some system parameters, which do not intervene in the definition of the strategies, are not perfectly known, and are hence better represented by distributions. These uncertain parameters are denoted by a vector $\bar{a}$, defined in a space $A \subset R^p$. Vector $\bar{a}$ includes e.g. component failure rates, random delays in control devices, magnitudes of initiating events ... These stochastic aspects of the problem lead to the definition of a probability function $P_\varphi(\bar{u})$ that the total cost does not exceed a maximal admissible budget φ:

$$P_\varphi(\bar{u}) = P\left\{ \bar{a} \in A \,|\, \phi(\bar{u},\bar{a}) \le \varphi \right\} \qquad (2)$$

The desired optimal strategy $\bar{u}_\varphi$ is that maximising the probability function

$$\bar{u}_\varphi = \underset{\bar{u} \in U}{\operatorname{argmax}} \, P_\varphi(\bar{u}) \qquad (3)$$

In stochastic programming (Kibzun & Kan 1996), eqns.(2) and (3) define a primal stochastic problem. We have implicitly assumed that potential constraints on the possible strategies are accounted for in the definition of U, what avoids supplementary complications in our problem.

A general iterative scheme of the form

$$\bar{u}_{k+1} = \Pi_U[\bar{u}_k + \rho_k \bar{\xi}_k] \qquad (4)$$

can be used to solve the primal stochastic problem. In eqn.(4), Π_U is the Euclidian projector on the space $U \subset R^m$, ρ_k is the magnitude of the change in strategy at this iteration, and $\bar{\xi}_k$ should represent the gradient of the probability function $P_\varphi(\bar{u})$ at point $\bar{u}_k$. However, the computation of the gradient is a time-consuming numerical operation: indeed, it implies the integration of the cost function of all valus of $\bar{a} \in A$, for several values of $\bar{u}$ in a neighbourhood of $\bar{u}_k$. Instead, (Kibzun & Kan 1996) propounds the use of a stochastic quasi-gradient, which is a random vector such that

$$E(\bar{\xi}_k|\bar{u}_k) = a(\bar{u}_k)\overline{\nabla}_{\bar{u}} P_\varphi(\bar{u}_k) + \bar{b}(\bar{u}_k) \qquad (5)$$

When $a(\bar{u}) = 1$ and $\bar{b}(\bar{u}) = 0$, the stochastic quasi-gradient is unbiased. This property is satisfied if we define $\bar{\xi}_k$ in the following way. Let $\hat{a}_1, \ldots, \hat{a}_t$ be t samples of the random parameters $\bar{a}$. An unbiased estimator of the probability function is given by

$$P_t^*(\bar{u}) = \frac{1}{t}\sum_{i=1}^{t} H(\varphi - \phi(\bar{u}, \hat{a}_i)) \qquad (6)$$

if $H(.)$ is the Heaviside stepfunction.

Consider a parameter δ_k at the k^{th} iteration, and let $\tilde{u}_{kj}$ represent random variables uniformly distributed on the intervals $[u_{kj} - \delta_k, u_{kj} + \delta_k], j = 1 \ldots m$. Then, the stochastic quasi-gradient $\bar{\xi}_k$ used in (Kibzun & Kan 1996), p.259-262, is given by

$$\begin{aligned} \bar{\xi}_k \;=\; & \frac{1}{2\delta_k}\sum_{j=1}^{m}[P_{t_k}^*(\tilde{u}_{k1}, \ldots, u_{kj} + \delta_k, \ldots, \tilde{u}_{km}) \\ & - P_{t_k}^*(\tilde{u}_{k1}, \ldots, u_{kj} - \delta_k, \ldots, \tilde{u}_{km})]\bar{e}_j \qquad (7) \end{aligned}$$

where $\bar{e}_j, j = 1 \ldots m$, define the canonical basis on R^m.

Then, $\bar{u}_k$ is shown to converge almost surely to the optimal $\bar{u}_\varphi$, if the iterative method embodied by eqns.(4) to (7) is used, provided the following conditions are fulfilled:

(i) the probability function $P_\varphi(\bar{u})$ is concave and Lipschitz on the convex, compact domain U; the optimum $\bar{u}_\varphi \in \mathrm{int}(U)$.

(ii) the (deterministic) sequence $\{\rho_k\}$ satisfies

$$\rho_k > 0\,, \sum_{k=1}^{\infty} \rho_k = \infty\,, \text{ and } \sum_{k=1}^{\infty} \rho_k^2 < \infty$$

(iii) the (deterministic) sequences $\{\delta_k\}$ and $\{t_k\}$ satisfy

$\delta_k > 0\,, \delta_k \to 0\,, t_k \to \infty$, and $k^{1+\epsilon}/(\delta_k^2 t_k) \to 0$ as $k \to \infty$, for some $\epsilon > 0$.

Each iteration of the algorithm described in the previous paragraph calls for running two nested loops of MC simulation. The inner loop corresponds to the estimation of the cost function $\phi(\bar{u}, \bar{a})$ for given values of $\bar{u}$ and $\bar{a}$, while the outer loop performs the samplings of the random vector $\bar{a}$. The latter issue is tackled by using Latin Hypercube Sampling, in order to achieve a sampling of domain A which is as systematic as possible.

The estimation of the cost function is a more complicated task, since it involves the estimation of the probability of system shutdown and that of system failure. These situations usually refer to rare combinations of events, which are unlikely to be often sampled in an analogue simulation. Therefore, we have implemented in the algorithm efficient MC techniques that were developed to assess small failure risks in dynamic reliability (Labeau 1996; Labeau 1997). Two free-flight estimators are used: one for the probability of a system shutdown, and the other one associated with failure situations. These estimators consist in scoring the probability of an event of interest (shutdown or failure) from each stage of an history, be this event sampled or not. These estimators are combined with the memorisation - before the simulation proper - of the most probable evolutions from any possible initial state the transient can start in. These trajectories correpond to the expected behaviour of the system after the initiating event, if all control and protection devices are correctly working, and if all transitions in operation are disabled. This allows to speed up the simulation, as well as to drive the histories towards unexpected sequences of events. The net result is a more accurate estimation of the probability of rare events.

3 APPLICATION

In order to illustrate the optimisation procedure described in section 2, we have applied it on a HNO_3 cooling system (Signoret et al. 1997). It has been previously used as a test case in order to assess the capabilities of risk assessment methods to deal with time-dependent processes (Pasquet et al. 1997), but no truly dynamic analysis had been performed on this example yet.

3.1 Description

The system under study is presented in fig.1. Its mission consists in cooling down a flow of nitric acid within a specified temperature range, the whole system being a supply unit in acid for an-

within the desired interval, the cooling system is equipped with two nested feedback loops. These are designed to modify the characteristics of the heat exchanger in order to adapt to variations in the input temperature. The inner loop comprises an automatic controller and a human operator, whose actions prevail on the controller's ones. The manual shutdown of the system, performed by the operator if the output temperature goes out of the required interval, constitutes the outer loop.

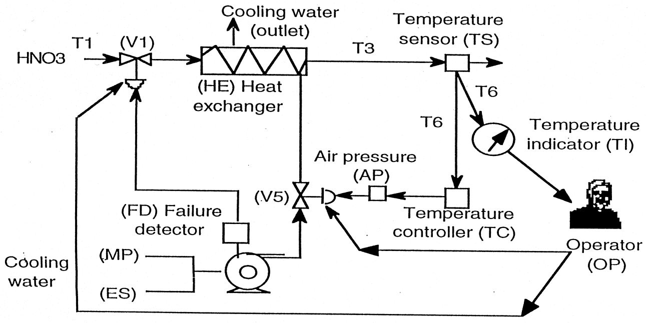

Figure 1: HNO_3 cooling system

The FMEA of the system is summarized in the following enumeration of the components, together with the various states they can lie in (Signoret et al. 1997):

1. valve $V1$. This is the inlet point of HNO_3 in the cooling system. $V1$ can be either open, closed (in case of shutdown), or stuck open.

2. temperature sensor $T1$. This device measures the initial temperature of nitric acid. It can be found in any of the following four states: operational, failed-stuck, or displaying a positive or a negative set-point deviation.

3. temperature sensor TS. It measures the final temperature of HNO_3, and has the same states as $T1$.

4. valve $V5$. This valve controls the cooling water flow. It is either working or stuck.

5. air pressure AP. This device is acting on $V5$ to control its position. It can become inefficient in case of air pressure loss.

6. main pressure MP. This is the main cooling water supply, which becomes unavailable in case of main pressure loss.

7. emergency supply ES. This secondary water supply has the same states as MP.

8. pump P. This pump causes a cooling water flow. It can be working or failed.

provides the only signal accessible to the operator OP, by displaying the output temperature of HNO_3. The same states as those for $T1$ are possible.

10. failure detector FD. Its mission consists in closing $V1$ if P fails, if it is not blocked.

11. operator OP. The operator has no rôle as long as the cooling system works as expected and the automatic controller is operational. But he comes into play as soon as critical values of TI are transmitted. He can be present, absent, worried or panicked. As we did not intend to implement a sophisticated human reliability model in order to better concentrate on the optimisation algorithm, the last two states were introduced, together with a delay before operator action. This allows for a basic modelling of a diagnostic time and of the ordering in the actions of the operator.

12. temperature controller TC. This device is intended to compute the optimal position of $V5$ according to the values given by $T1$ and TS. It displays the same states as $T1$.

The heat exchanger itself is assumed to stay operational during the whole transient.

This cooling system displays two major characteristics: it is sequential and non coherent (Signoret et al. 1997). Therefore, the timing and ordering of events strongly influence the consequences of accident sequences. A dynamic approach to this risk analysis problem is thus well adapted (Siu 1994). But up to now, no dynamic model of the system had been provided. Evolution equations are to be written for the main process variables describing the cooling installation, for all configurations it can be found in. The full expression of these dynamics can be found in (Bourgeois & Labeau 1999), while the set of process variables that are taken into account is given herebelow:

1. input temperature of HNO_3. This variable obeys a given increasing function of time, which models the initiating event causing the transient in the system.

2. cooling water flow. As long as the automatic controller is operational, the value of this flow exponentially converges to the optimal water flow computed by TC, given the desired output temperature of HNO_3. Its variation only depends on TI if OP is solicitated. If AP or $V5$ is stuck, it remains constant.

3. output temperature of HNO_3. Its value is calculated using the heat exchanger properties, for given values of the input temperature of HNO_3 and the water flow.

ator's next action, it remains constant unless OP is worried or panicked.

5. value of $T1$. This indication is supposed to be equal to the input temperature of acid, though it could be constant or biased, depending on the state of $T1$.

6. value of TS. This variable plays the same rôle as the value of $T1$ for the output temperature.

7. value used in TC. This temperature indication is equal to the value of TS, plus or minus a bias in case of setpoint deviation. It is constant if TC is stuck.

8. value of TI. This is the information accessible to OP, and its behaviour is similar to that of the previous variable.

9. optimal water flow for TC. It expresses the cooling water flow corresponding to the desired output temperature of acid, provided the input temperature communicated to TC is the actual one.

10. computed water flow for TC. This variable gives the water flow leading to an output temperature equal to that received by TC, given the value of $T1$.

We still have to define the system failure: it occurs either because the output temperature leaves an admissible range, or because an unacceptable hardware state is entered. The latter situation corresponds in our case to the loss of the cooling water flow, when MP and ES are simultaneously failed, or when P fails while $V1$ is still open. Let us finally remind that the system can be shut down by the operator, by manually closing $V1$.

3.2 The primal stochastic problem

After having described our cooling system, its possible component states and the dynamic model of the system evolution following a variation in the inlet temperature, we now have to specify the optimisation problem itself. We consider a bidimensional vector of strategies. The first component of $\bar{u}$ is associated with the quality of the temperature sensors to be used in the system. This characteristic is represented by the value of the setpoint deviation they are likely to display. The choice of u_2 is related to the operator's mission, through a desired probability of presence (and hence of undelayed reaction to a potential transient). The variations of u_2 then corresponds to variations in the value of the transition rate between the "present" and "absent" states OP can be in.

for in vector $\bar{a}$: a_1 gives the variations of the failure rates of the various temperature sensors, while a_2 determines the characteristics of the initiating event (magnitude and rise time of the inlet temperature modification).

Let X be one of the physical quantities influenced by the strategies. We choose to express X as a linear function of its extreme values X_o and X_1, and of u_1 or u_2:

$$X = X_o + u_i(X_1 - X_o) \qquad (8)$$

where $i = 1$ or $i = 2$. The same treatment is done for the distributed parameters described by $\bar{a}$. Therefore, both vectors $\bar{u}$ and $\bar{a}$ are defined on the domain $[0,1] \times [0,1]$. For the sake of simplicity, we assume the total cost linearly depends on the vector of strategies, so that eqn.(1) becomes:

$$\phi = \bar{c}^T.\bar{u} + p_s\phi_s + p_f\phi_f \qquad (9)$$

where $\bar{c}$ is a constant vector.

3.3 Numerical tests

The algorithm presented in eqns.(4) to (7) was applied to the nitric acid cooling system, while considering the cost function (9). The results obtained with this procedure turned out to be unsatisfactory. This is mainly due to the nature of the estimator (6) of $P_\varphi(\bar{u})$, and its use in the calculation of the stochastic quasi-gradient (7). Indeed, two strategies belonging to the same neighbourhood of $\bar{u}_k$ are likely to give, for each sample of $\bar{a}$, two values of ϕ which are both larger (or smaller) than φ. But this situation leads to an estimation of the quasi-gradient that vanishes in many cases, because of the stepfunctions in the expression of $P_t^*(\bar{u})$. This makes the algorithm totally inefficient.

In order to address this problem, it is worth reminding that the cost function is estimated via a MC simulation computing p_s and p_f. This MC game provides us with an estimation $\phi(\bar{u}, \bar{a})$ of the cost function, which is the expected value of the random variable $\Phi(\bar{u}, \bar{a})$ associated with the MC algorithm. This random variable has a normal distribution, because of the central-limit theorem, and its variance can be estimated during the same simulation. Therefore, we can replace the estimator (6) with the following expression :

$$P_t^*(\bar{u}) = \frac{1}{t}\sum_{i=1}^{t} \text{Prob}(\Phi(\bar{u}, \hat{a}_i) < \varphi) \qquad (10)$$

With this new estimator, the optimisation algorithm displays a more interesting behaviour. It has thus been used in the sequel of the numerical tests.

An important observation was done when studying the optimisation procedure on our test-case: termining the success or failure of the algorithm. Indeed, an overestimated value of φ leads to values of $P_\varphi(\bar{u})$ which are very close to 1, whatever strategy is selected. No distinction between values of $\bar{u}$ can then be achieved. On the other hand, a small value of φ entails an erratic behaviour of the algorithm, as most estimations of ϕ exceed the budget φ. This undesired situation is illustrated by the evolution of $P_\varphi(\bar{u})$ during the first iterations of the procedure, as displayed in table 2, if the parameters of the optimisation problem take the values given in table 1. Table 2 also gives the evolution of ϕ_{avg}, average cost on all estimations performed with the current strategy. Obviously, as most estimated costs are larger than φ, the algorithm tends to select strategies enlarging the dispersion of $\phi(\bar{u}, \bar{a})$ to maximise $P_\varphi(\bar{u})$. But such a situation is unacceptable, because it drives the optimisation scheme towards strategies likely to give important values of ϕ for specific values of $\bar{a}$.

Table 1: Parameters of the optimisation problem

c_1	c_2	ϕ_f	ϕ_s	φ
1.0	3.0	1000.0	10.0	5.4

Table 2: Evolution of the probability function and of the average cost

iter	u_1	u_2	ϕ_{avg}	P_φ
0	0.60	0.33	8.32 ± 4.68	0.227
1	0.92	0.01	12.89 ± 8.82	0.235
2	0.98	0.01	13.87 ± 8.55	0.267
3	0.98	0.01	14.09 ± 8.52	0.258
4	0.98	0.13	14.32 ± 8.58	0.243

To obtain a relevant value of φ, one can use the estimations of $\phi(\bar{u}, \bar{a})$ realised with the initial strategy $\bar{u}_o$. Numerical experiments have shown that a value of φ slightly larger than the median of the computed costs obtained with $\bar{u}_o$ was a good start for the algorithm. But after some iterations, the current strategy can give values of ϕ for which the initial budget φ is not convenient anymore. To tackle this new difficulty, we propound the use of an adaptable budget: when the performances of the algorithm become unsatisfactory with the first guess for φ, a new value φ' can be determined, based on the costs obtained with the current strategy. This operation defines a new optimisation problem, which leads to strategies associated with a smaller ϕ_{avg}. This situation is illustrated in tables 3 and 4.

Table 3: Parameters of the optimisation problem

c_1	c_2	φ_f	φ_s	φ	φ
2.0	1.0	300.0	10.0	2.9	2.56

Table 4: Evolution of the probability function and of the average cost

iter	u_1	u_2	ϕ_{avg}	P_φ
0	0.50	0.50	2.67 ± 0.48	0.662
1	0.64	0.48	2.30 ± 0.52	1.000
1'				0.599
2	0.83	0.50	2.07 ± 0.90	0.950

Finally, we have performed a sensitivity analysis on the choice of the initial strategy. The values of $\bar{u}$ obtained after a few iterations clearly depend on $\bar{u}_o$. An "optimal region" belonging to U is underlined in this fashion, and a set of equivalent strategies associated with values of P_φ very close to each other is found by repeating the calculations for several values of $\bar{u}_o$. Even if our algorithm does not appear capable of finding its way in this optimal zone, the propounded sensitivity analysis can provide a map of this region, that can help the designer to select the most appropriate strategy.

4 SUMMARY AND CONCLUSIONS

This work is dedicated to the implementation of stochastic programming techniques within a dynamic PRA context. Such a coupling can be envisioned to define an optimal design of protection devices, or to select the optimal values of parameters of emergency procedures.

The reasons why we had to resort to stochastic programming were first given: the total cost, including the investment necessary to implement a given strategy as well as the cost or loss of revenue entailed by a system failure or its shutdown, is a random function, due to the presence of distributed parameters in the risk analysis. Rather than minimising the expected value of this total cost, we maximise, in our reliability context, the probability that the total cost does not exceed a critical value. This way, our optimisation problem takes the form of a primal stochastic problem.

An algorithm based on the estimation of a stochastic quasi-gradient of the probability function was considered to obtain the optimal strategy. It was applied to a HNO_3 cooling system, the dynamic description of which was realised. The numerical tests have demonstrated that: a. the estimator (6) of $P_\varphi(\bar{u})$ is inefficient; b. the choice of the maximal budget φ is critical; c. the initial strategy in the optimisation scheme influences the convergence of the algorithm.

erence method which were propounded to tackle these issues have led to significant improvements in the search for the optimal strategy. Yet the convergence of the scheme still appears unsatisfactory when the strategy enters an optimal zone. Various reasons can explain this undesired behaviour : a. the probability function in our problem is not convex, as required to ensure the convergence of the algorithm; b. the definition of the cost function does not allow a sufficiently fine selection between neighbouring strategies ...

Among possible perspectives for future developments of the method, let us cite the use of another optimisation method (like simulated annealing) within an optimal zone, a non linear parameterisation of the strategies, or the expression of the optimisation problem as an inverse stochastic problem (Kibzun & Kan 1996). In the latter case, one aims at minimising the critical value φ of the total cost such that $P_\varphi(\bar{u})$ exceeds a desired level of confidence. Such approach would allow to get rid of the difficulties related to the choice of φ.

REFERENCES

Bourgeois, F. & Labeau, P. E. (1999). Stochastic quasi-gradient based optimization algorithms for dynamic reliability calculations. submitted for publication in Reliability Engineering and System Safety.

Devooght, J. (1997). Dynamic reliability. Advances in Nuclear Science and Technology 25, 215–278.

Kibzun, A. & Kan, Y. (1996). Stochastic programming problems with probability and quantile functions. Wiley-Interscience in systems and optimization. Chichester: John Wiley & Sons.

Labeau, P. E. (1996). Probabilistic dynamics: estimation of generalized unreliability through efficient Monte Carlo simulation. Annals of Nuclear Energy 23:17, 1355–1369.

Labeau, P. E. (1997). Variance reduction techniques in Monte Carlo simulation applied to dynamic reliability. In C. G. Soares (Ed.), Proceedings of Esrel'99, Lisboa, pp. 2129–2137. Pergamon.

Pasquet, S., Châtelet, E., Thomas, P., & Dutuit, Y. (1997). Analysis of a sequential non coherent and looped system with two approaches: Petri nets and neural networks. In C. G. Soares (Ed.), Proceedings of Esrel'99, Lisboa, pp. 2257–2263. Pergamon.

Signoret, J. P., Harvey, D., Dutuit, Y., & Châtelet, E. (1997). Test-case activity. Lisboa: ESRA-ISdF. Report from the Esrel'97 workshop.

Siu, N. (1994). Risk assessment for dynamic systems: an overview. Reliability Engineering and System Safety 43, 43–73.