Active learning surrogate models for the conception of systems with multiple failure modes

G Perrin

Active learning surrogate models for the conception of systems with multiple failure modes

G. Perrin[a]

[a] *CEA/DAM/DIF, F-91297, Arpajon, France*

Abstract

Due to performance and certification criteria, complex mechanical systems have to take into account several constraints, which can be associated with a series of performance functions. Different software are generally used to evaluate such functions, whose computational cost can vary a lot. In conception or reliability analysis, we thus are interested in the identification of the boundaries of the domain where all these constraints are satisfied, at the minimal total computational cost. To this end, the present work proposes an iterative method to maximize the knowledge about these limits while trying to minimize the required number of evaluations of each performance function. This method is based first on Gaussian process surrogate models that are defined on nested sub-spaces, and second, on an original selection criterion that takes into account the computational cost associated with each performance function. After presenting the theoretical basis of this approach, this paper compares its efficiency to alternative methods on an example.

Keywords:
Computer experiments, Gaussian Processes, system reliability, sequential design

1. Introduction

The conception (or risk assessment) of complex mechanical systems has to take into account a series of constraints. Such constraints can be due to certification criteria, performance objectives, cost limitations, and so on. In this context, the role of simulation has kept increasing for the last decades, as

Email address: guillaume.perrin2@cea.fr (G. Perrin)

one should be able to predict if a given configuration of the system is likely to fulfil these constraints without having to build it and to test it experimentally. In many cases, the computation of these constraints is associated with a series of computer software, whose physics can vary a lot. For instance, in the car industry, the conception of a new vehicle can be subjected to constraints on its size and weight, which are rather easy to compute, but also on its emergency stopping distance, its crash or aerodynamic resistance, which can be much more difficult to evaluate.

To be more precise, let us consider a particular system, $\mathcal{S}$, which design is supposed to be characterized by a vector of d parameters, $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$. It is assumed that the system constraints can be evaluated from the computation of $N \geq 1$ performance functions, $\{g_n,\ 1 \leq n \leq N\}$, which respective numerical cost (in CPU time for instance), C_n, are supposed to be sorted in an ascending order :

$$C_1 \leq C_2 \leq \cdots \leq C_N. \tag{1}$$

Thus, the conception domain, which is denoted by Ω and which defines the set of admissible designs for the considered system, can be written as :

$$\Omega = \bigcap_{n=1}^{N} \Omega_n, \quad \Omega_n = \left\{ \boldsymbol{x} \in \mathbb{R}^d, \quad g_n(\boldsymbol{x}) \leq 0 \right\}. \tag{2}$$

Such a domain is a key element to perform optimizations of the system restricted to admissible design solutions, while being closely linked to reliability analysis prospects, as its complementary, $\mathbb{R}^d \backslash \Omega$, corresponds to the failure domain of the system. Hence, for the last decades, the identification of Ω, or of its boundary, $\partial\Omega$, has motivated the development of several methods, which can be sorted in two main categories : the direct and the indirect methods. Among the direct methods, the first-order or second-order reliability methods (FORM/SORM) approximate $\partial\Omega$ as a linear or a second-order polynomial function [9, 14, 13, 4]. When confronted to applications where the limit state is multimodal or is strongly non-linear, alternative methods based on more advanced approximations have been introduced, such as support vector machines (SVM) techniques [19, 17, 11] and methods based on generalized least-squares linear regression [18, 10].

On the other hand, the indirect methods focus on the approximation of the performance functions to deduce in a second step the searched boundary. Among these methods, the Gaussian process regression (GPR) method, or

kriging, keeps playing a major role, which is due to its ability to provide a robust approximation of $\partial\Omega$, that is to say for which precision can be quantified [15, 12, 16, 7].

Based on this very efficient tool, the idea of this paper is to present a sequential sampling strategy to minimize the uncertainties about boundary $\partial\Omega$, at the minimal computational budget. In particular, the proposed strategy will take into account the computational costs associated with the evaluation of each function, $\{C_1, \ldots, C_N\}$.

The outline of this work is as follows. First, Section 2 presents the theoretical bases of the Gaussian process regression (GPR) and its use for the identification of limit states. The proposed method is then introduced in Section 3. Then, the efficiency of the method is illustrated on an analytic example in Section 4.

2. Surrogate models for system reliability

The Gaussian process regression is based on the assumption that each performance function, g_n, $1 \leq n \leq N$, can be seen as a sample path of a stochastic process, which is supposed to be Gaussian for the sake of tractability. By conditioning this Gaussian process by a set of $Q \geq 1$ code evaluations, $\mathcal{S}^{\text{learn}} = \left\{(\boldsymbol{x}^{(q)}, g_n(\boldsymbol{x}^{(q)})),\ 1 \leq q \leq Q\right\}$, it is possible to define very interesting predictors for the value of g_n in any non-computed point of the input space. These predictors of functions g_n at any $\boldsymbol{x}$ in $\mathbb{R}^d$, which are respectively denoted by $\widehat{g}_n(\boldsymbol{x})$, are Gaussian by construction, $\widehat{g}_n(\boldsymbol{x}) \sim \mathcal{N}(\widehat{\mu}_n(\boldsymbol{x}), \widehat{\sigma}_n^2(\boldsymbol{x}))$, and we refer to [15, 16] for further details about the expression of the conditioned means, $\widehat{\mu}_n(\boldsymbol{x})$, and standard deviations, $\widehat{\sigma}_n^2(\boldsymbol{x})$. Such a predictor interpolates in the sense that, for all $1 \leq q \leq Q$,

$$\mathbb{P}\left(\widehat{g}_n(\boldsymbol{x}^{(q)}) = g_n(\boldsymbol{x}^{(q)})\right) = 1. \tag{3}$$

It is moreover sequentially improved : for all $\boldsymbol{x}$ in $\mathbb{R}^d$, the higher is Q, the smaller the integrated mean square error (IMSE), $\mathbb{E}\left[\int_{\mathbb{R}^d}(g_n(\boldsymbol{x}) - \widehat{g}_n(\boldsymbol{x}))^2 d\boldsymbol{x}\right]$, is supposed to be. Under the assumption that $\widehat{\mu}_n$ is a good predictor of g_n, a good approximation of $\partial\Omega$ is therefore given by the elements of $\mathbb{R}^d$ such that $\mathbb{P}(\widehat{g}_n(\boldsymbol{x}) \leq 0) = \mathbb{P}(\widehat{g}_n(\boldsymbol{x}) \geq 0) = 1/2$, which yields :

$$\partial\Omega_n \approx \widehat{\partial\Omega}_n = \left\{\boldsymbol{x} \in \mathbb{R}^d,\ \ \widehat{\mu}_n(\boldsymbol{x}) = 0\right\}. \tag{4}$$

Function $\widehat{\sigma}_n$ can then be used to quantify the precision of such an approximation $\widehat{\partial\Omega}_n$, as the smaller $\widehat{\sigma}_n(\boldsymbol{x})$ is, the more chance there is for $g_n(\boldsymbol{x})$ and $\widehat{\mu}_n(\boldsymbol{x})$ to be close. Improving the knowledge about $\widehat{\partial\Omega}_n$ amounts therefore at adding new points $\boldsymbol{x}^{(n),\star}$ to the learning set $\mathcal{S}^{\text{learn}}$, which have to be chosen according to a specific criterion to minimize the total computational cost for a given precision. Such new points are generally chosen iteratively such that :

$$\boldsymbol{x}^{(n),\star} = \arg \max_{\boldsymbol{x}\in\mathbb{R}^d,\ \ \widehat{\mu}_n(\boldsymbol{x})=0} \widehat{\sigma}_n(\boldsymbol{x}), \tag{5}$$

that is to say where the expected value of g_n is the closest to the threshold ($\widehat{\mu}_n(\boldsymbol{x}) = 0$) with the largest uncertainty. Solving the problem defined by Eq. (5) being complex, two adaptations have been proposed to provide a balance between exploration and exploitation. On the first hand, the Efficient Global Reliability Analysis (EGRA) method (see [3] for further details) replaces such a constrained maximization of the standard deviation $\widehat{\sigma}_n$, by the unconstrained maximization of a learning function called Expected Feasibility Function, EF, which writes :

$$\text{EF}(\boldsymbol{x}, n) = \mathbb{E}\left[\epsilon(\boldsymbol{x}) - \min\left(|\widehat{g}_n(\boldsymbol{x})|, \epsilon(\boldsymbol{x})\right)\right], \tag{6}$$

where ϵ is a function chosen to focus the search in the immediate vicinity of $\partial\Omega_n$ (for instance, ϵ can be chosen proportional to $\widehat{\sigma}_n$). Details on the implementation and the maximization of function EF can be found in [3]. On the other hand, the Active learning and Kriging-based Monte-Carlo Simulation (AK-MCS) method (see [6]) proposes a discrete adaptation of the optimization problem defined by Eq. (5) :

$$\boldsymbol{x}^{(n),\star} \approx \arg \min_{\boldsymbol{z}\in\left\{\boldsymbol{z}^{(1)},\ldots,\boldsymbol{z}^{(\nu)}\right\}} \frac{|\widehat{\mu}_n(\boldsymbol{z})|}{\widehat{\sigma}_n(\boldsymbol{z})}, \tag{7}$$

where $\left\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(\nu)}\right\}$ is a set of ν vectors that are randomly chosen in $\mathbb{R}^d$. Therefore, both former methods realize a trade-off between exploration of each boundary $\partial\Omega_n$ of Ω_n and global uncertainty reduction, at a relatively small numerical cost.

When interested in identifying the boundary of the intersection of domains $(\Omega_n)_{1\leq n\leq N}$, a very simple strategy would be to use either the EGRA or the AK-MCS iterative method to train each model g_n to sufficient accuracy, and then identify space $\Omega = \Omega_1 \cap \cdots \cap \Omega_N$, as :

$$\Omega = \left\{ \boldsymbol{x} \in \mathbb{R}^d, \quad \max_{1 \leq n \leq N} g_n(\boldsymbol{x}) \leq 0 \right\} \approx \left\{ \boldsymbol{x} \in \mathbb{R}^d, \quad \max_{1 \leq n \leq N} \widehat{\mu}_n(\boldsymbol{x}) \leq 0 \right\}. \quad (8)$$

However, it is clear that such a procedure can lead to many useless evaluations of the performance functions. For instance, if $\Omega_1 \subset \Omega_2$, no evaluations of g_2 are needed to analyse the boundary of Ω. To limit the number of calls to performance functions that have little or no influence on the definition of $\partial\Omega$, it would seem interesting to directly apply the EGRA or the AK-MCS methods to the composite function $g^{\max} = \max_{1 \leq n \leq N} g_n$. However, it appears that such an approach is affected by several problems. Indeed, even if functions g_n are regular, $g^{\max}$ is generally highly irregular and its modeling by a GPR- based surrogate model can be difficult and lead to additional expense. Hence, instead of working on the aggregation of the performance functions, it appears to be more efficient to still consider the approximations of each performance function g_n by a GPR-based surrogate model, and choose a selection criterion that is adapted to the system case. To this end, let $n^\star$ be the index such that, for all $\boldsymbol{x}$ in $\mathbb{R}^d$,

$$n^\star(\boldsymbol{x}) = \arg \max_{1 \leq n \leq N} \widehat{\mu}_n(\boldsymbol{x}). \quad (9)$$

Therefore, it has been proposed in [2] and [8] to choose the new evaluation point, $\boldsymbol{x}^\star$, such that $\boldsymbol{x} \mapsto \mathrm{EF}(\boldsymbol{x}, n^\star(\boldsymbol{x}))$ is maximal, or such that :

$$\boldsymbol{x}^\star = \arg \min_{\boldsymbol{z} \in \left\{ \boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(\nu)} \right\}} \frac{|\widehat{\mu}_{n^\star(\boldsymbol{z})}(\boldsymbol{z})|}{\widehat{\sigma}_{n^\star(\boldsymbol{z})}(\boldsymbol{z})}, \quad (10)$$

to accurately adapt the EGRA and the AK-MCS procedures to the system case, respectively. At this new point $\boldsymbol{x}^\star$, only the true performance function, $\boldsymbol{x} \mapsto g_{n^\star(\boldsymbol{x})}(\boldsymbol{x})$, has to be computed, such that only a small number of calls to true performance functions that have little influence on $\partial\Omega$ should be made.

It can be noticed that such pointwise strategies do not take into account in their selection criteria the fact that the new evaluation point will bring additional information on its neighbourhood. In contract, Stepwise Uncertainty Reduction (SUR) approaches [1] propose to choose the new evaluation point in order to minimize the expected value of a well chosen measure of the uncertainty about the search domain. For instance, if we denote by $\mathbb{V}(\Omega, m)$

the variance of the volume of Ω, which is conditioned by all the available code evaluations at step m, then the new point, $\boldsymbol{x}^*$, can be chosen such that :

$$(\boldsymbol{x}^*, n^*) = \arg \min_{\boldsymbol{x}^{m+1} \in \mathbb{R}^d, 1 \leq n \leq N} \mathbb{E}\left[\mathbb{V}(\Omega, m+1) \mid g_n(\boldsymbol{x}^{m+1}) = \widehat{g}_n(\boldsymbol{x}^{m+1})\right]. \quad (11)$$

Such methods based on global measures of uncertainty of Ω have been shown to outperform pointwise approaches for the identification of excursion set on a series of applications based on a single performance function[1]. In spite of recent algorithmic developments [5], the main drawback of these methods is the fact that the solving of Eq. (11) is very computer demanding, even if only one performance function is considered ($N = 1$). This explains why such promising approaches have, to the author knowledge, almost never been applied to applications where the performance of the system is characterized by more than one function. As a consequence, only the pointwise approaches, which are more flexible and less time consuming, will be considered in the following.

3. Nested GPR-based surrogate models for system conception and reliability analysis

In the adaptations of the EGRA and of the AK-MCS procedures, no attention is paid to the computational cost associated with each performance function g_n. Moreover, the definition domains of each surrogate model, $\widehat{g}_n$, are the same (the entire space for EGRA or a finite set of input candidates for AK-SYS), which can be overly expensive. This motivates the introduction of a new adaptation for the approximation of $\partial\Omega$ using GPR-based surrogate models. The idea of this adaptation is to limit as much as possible the number of calls to the performance functions which computational costs are the highest. To this end, it is first proposed to limit the approximations of functions g_n on nested sub-spaces of $\mathbb{R}^d$, and then to take into account the computational costs of the performance functions in the iterative selection criterion. Indeed, keeping in mind that the computational costs of the performance functions are sorted in an increasing order, $C_1 \leq \ldots, \leq C_N$, and noticing that :

$$\mathbb{P}(\boldsymbol{x} \in \Omega) = \mathbb{P}(\boldsymbol{x} \in \Omega_N \mid \boldsymbol{x} \in \bigcap_{k=0}^{N-1} \Omega_k) \times \cdots \times \mathbb{P}(\boldsymbol{x} \in \Omega_2 \mid \boldsymbol{x} \in \Omega_1) \times \mathbb{P}(\boldsymbol{x} \in \Omega_1), \tag{12}$$

it seems interesting to focus the relevance of the predictions of functions g_n on the vicinity of $\bigcap_{k=0}^{n-1} \Omega_k$, with $\Omega_0 = \mathbb{R}^d$. To do so, from an initial design of experiments, $\mathcal{S}_1^{\text{learn}} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(Q)}\right\}$, we denote by $\widetilde{g}_n(\boldsymbol{x}) \sim \mathcal{N}(\widetilde{\mu}_n(\boldsymbol{x}), \widetilde{\sigma}_n^2(\boldsymbol{x}))$ the GPR-based approximation of g_n, which is computed according to the Algorithm 1, where for all $1 \leq n \leq N$, $\text{GPR}(g_n, \mathcal{S}_n^{\text{learn}})$ denotes the GPR-based surrogate model associated with the evaluations of g_n at the points gathered in $\mathcal{S}_n^{\text{learn}}$.

Initialization : evaluate g_1 at each position $\boldsymbol{x} \in \mathcal{S}_1^{\text{learn}}$ and compute $\widetilde{g}_1 = \text{GPR}(g_1,\ \mathcal{S}_1^{\text{learn}})$;
for $2 \leq n \leq N$ **do**
  extract $\mathcal{S}_n^{\text{learn}} = \left\{\boldsymbol{x} \in \mathcal{S}_{n-1}^{\text{learn}} \mid g_k(\boldsymbol{x}) \leq 0,\ 1 \leq k \leq n-1\right\}$;
  evaluate g_n at each position $\boldsymbol{x} \in \mathcal{S}_n^{\text{learn}}$;
  compute $\widetilde{g}_n = \text{GPR}(g_n, \mathcal{S}_n^{\text{learn}})$.
end

Algorithm 1: Definition of the conditioned GPR-based surrogate models.

GPR-based surrogate models being good at interpolating and rather bad at extrapolating, we deduce that $\prod_{n=1}^{N} \mathbb{P}(\widetilde{g}_n(\boldsymbol{x}) \leq 0)$ is a good estimation of $\mathbb{P}(\boldsymbol{x} \in \Omega)$ in the vicinity of Ω, and decreases relatively quickly with respect to the distance between $\boldsymbol{x}$ and Ω. It comes :

$$\partial\Omega \approx \widetilde{\partial\Omega} = \left\{ \boldsymbol{x} \in \mathbb{R}^d,\ \ \prod_{n=1}^{N} \mathbb{P}(\widetilde{g}_n(\boldsymbol{x}) \leq 0) = 1/2 \right\}. \tag{13}$$

The computational cost associated with such an approximation of $\partial\Omega$ thus is $\sum_{n=1}^{N} C_n \times \#\mathcal{S}_n^{\text{learn}}$, which has to be compared to $Q \times \sum_{n=1}^{N} C_n$, where $\#\mathcal{S}_n^{\text{learn}}$ denotes the cardinal of $\mathcal{S}_n^{\text{learn}}$ and Q is the dimension of the initial learning set. Hence, thanks to this nested structure, most of the evaluations are made on the cheapest performance functions, as their associated GPR-based surrogate models are defined in bigger spaces, whereas a limited number of

calls are made to the most expensive ones, as their associated GPR-based surrogate models have to be relevant in smaller spaces.

In the same manner than in Section 2, the precision of such a boundary can iteratively be improved by adding new evaluations of true performance functions $g_1, \ldots, g_N$ to the learning sets associated with the computation of each predictor $\widetilde{g}_n$. Once again, to limit the number of calls to the most expensive performance functions, it is proposed to evaluate function g_n at position $\boldsymbol{x}$ if and only if, for all $k < n$, $\mathbb{P}(\boldsymbol{x} \in \Omega_k) \geq P_k^\star$, where $P_k^\star$ is a probability threshold ($P_k^\star = 0.95$ for instance). As a consequence, the *expected* computational cost, $\mathcal{C}_n(\boldsymbol{x})$ associated with the evaluation of g_n in $\boldsymbol{x}$ is given by :

$$\mathcal{C}_n(\boldsymbol{x}) = C_1 \times \mathbb{I}_{\Omega_1}(\boldsymbol{x}) + \sum_{k=2}^{n} C_k \times \mathbb{I}_{\Omega_k}(\boldsymbol{x}) \times \prod_{u=1}^{k-1} \mathbb{P}(\widetilde{g}_u(\boldsymbol{x}) \leq 0), \tag{14}$$

$$\mathbb{I}_{\Omega_k}(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad \mathbb{P}(\widetilde{g}_k(\boldsymbol{x}) \leq 0) \leq P^\star, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

As a compromise between computational cost and uncertainty reduction, the new point $\boldsymbol{x}^\star$ is then chosen as the solution of the following optimization problem :

$$(n^\star, \boldsymbol{x}^\star) = \arg \max_{1 \leq n \leq N, \quad \boldsymbol{x} \in \widetilde{\partial\Omega}} \frac{\widetilde{\sigma}_n(\boldsymbol{x})}{C_n(\boldsymbol{x})}. \tag{16}$$

As for the AK-SYS method, the continuous search for the maximization of Eq. (16) is replaced in the following by a discrete search over a set of input candidates $\left\{\boldsymbol{x}^{(m)} \in \mathbb{R}^d,\ 1 \leq m \leq M\right\}$. As functions C_n, $\widetilde{\mu}_n$ and $\widetilde{\sigma}_n$ are cheap to evaluate, the value of M can be chosen high (up to several millions), which limits the sub-optimality of the discrete approach. Once position $\boldsymbol{x}^\star$ and index $n^\star$ have been identified, the Algorithm 2 can be used to update some of the predictors.

Finally, thanks to the proposed selection criterion and to the fact that the surrogate models are defined on nested sub-spaces of $\mathbb{R}^d$, only few evaluations will be made to true performance functions that have little influence on $\partial\Omega$, and most of these evaluations will be made to the performance functions whose computational costs are the lowest.

```
for 1 ≤ n ≤ n⋆ do
    if ℙ(g̃_n(x⋆) ≤ 0) ≤ P_k⋆ then
        evaluate g_n(x⋆) ;
        update g̃_n ;
        if g_n(x⋆) > 0 then
            break loop for.
        end
    end
end
```

Algorithm 2: Adaptive updating of the surrogate models.

4. Application

To illustrate the advantages of the approach presented in Section 3, let us consider the following problem, which is defined by the $N = 3$ performance functions g_1, g_2, g_3, such that for all $\boldsymbol{x} = (x_1, x_2)$ in $[0,1]^2$:

$$\left\{\begin{array}{l} g_1(\boldsymbol{x}) = \left(\dfrac{(x_1-0.6)^2}{0.1^2} - \dfrac{(x_2-0.5)^2}{0.15^2}\right)\left(\dfrac{(x_1-0.3)^2}{0.45^2} + \dfrac{(x_2-0.5)^2}{0.4^2}\right) - 1, \\ g_2(\boldsymbol{x}) = \dfrac{(x_1-0.55)^2}{0.35^2} + \dfrac{(x_2-0.5)^2}{0.2^2} + \dfrac{(x_1-0.55)(x_2-0.5)}{0.3^2} - 1, \\ g_3(\boldsymbol{x}) = \left\{\begin{array}{l} \mathbb{I}_1(\boldsymbol{x}) \times \left(\dfrac{(x_1-0.7)^2}{0.2^2} + \dfrac{(x_2-0.7)^2}{0.15^2} - \dfrac{(x_1-0.7)(x_2-0.7)}{0.2^2}\right) \\ + \mathbb{I}_2(\boldsymbol{x}) \times \left(\dfrac{(x_1-0.2)^2}{0.2^2} + \dfrac{(x_2-0.4)^2}{0.25^2}\right) - 1 \end{array}\right\}, \end{array}\right. \tag{17}$$

where :

$$\mathbb{I}_1(\boldsymbol{x}) = \begin{cases} 1 \text{ if } (x_1-0.7)^2 + (x_2-0.7)^2 \leq (x_1-0.2)^2 + (x_2-0.4)^2, \\ 0 \text{ otherwise,} \end{cases} \tag{18}$$

$$\mathbb{I}_2(\boldsymbol{x}) = \begin{cases} 1 \text{ if } (x_1-0.7)^2 + (x_2-0.7)^2 \geq (x_1-0.2)^2 + (x_2-0.4)^2, \\ 0 \text{ otherwise,} \end{cases} \tag{19}$$

The domain

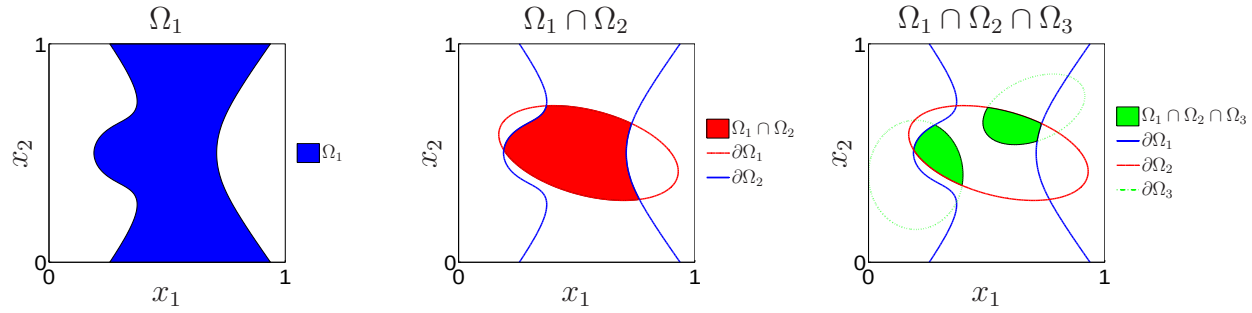

FIGURE 1: Representations of the domains Ω_1, $\Omega_1 \cap \Omega_2$ and $\Omega_1 \cap \Omega_2 \cap \Omega_3$.

$$\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3, \tag{20}$$

associated with these three functions, with $\Omega_n = \{\boldsymbol{x} \in [0,1] \times [0,1],\ g_n(\boldsymbol{x}) \leq 0\}$, is shown in Figure 1. Such a domain has been chosen, on purpose, neither convex nor connected.

As the estimation of the boundary $\partial\Omega$ of Ω is a goal that is closely related to the estimation of the probability $\mathbb{P}(\boldsymbol{x} \in \Omega)$, any procedure proposed for one of these objectives is expected to perform reasonably well for the other one. Therefore, this section compares the AK-MCS procedure (the EGRA approach giving the same results on this example) and the proposed approach, to identify the best estimation of $\partial\Omega$. Of course, a different performance ranking could have been obtained if the probability assessment point of view had been chosen.

In this prospect, we denote by $\%\mathrm{Vol}_{0.95}(\Omega)$ the ratio between the volume of the domain

$$\left\{\boldsymbol{x} \in [0,1]^2,\ \mathbb{P}(\boldsymbol{x} \in \Omega) \geq 0.95\right\}, \tag{21}$$

and the volume of Ω. From a conception (or reliability) point of view, we are therefore interested in the method that could allow the maximization of $\%\mathrm{Vol}_{0.95}(\Omega)$ at the minimal total computational cost.

4.1. Interest of the nested structure

At first, we suppose that the computational costs, C_1, C_2, C_3, which are associated with the three performance functions g_1, g_2, g_3 respectively, are

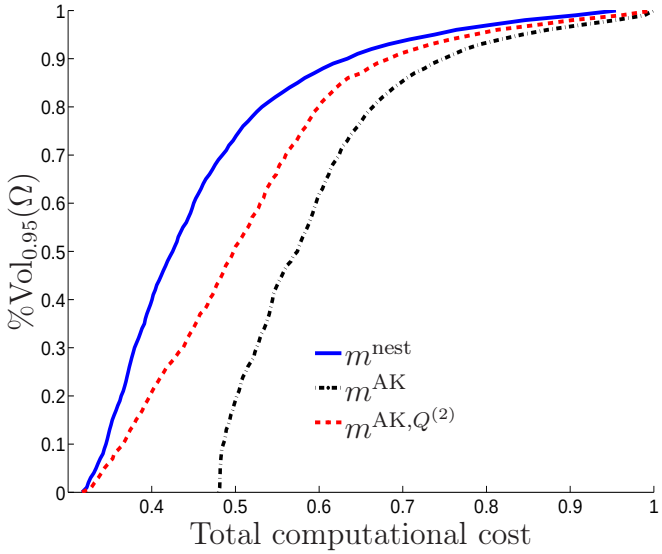

FIGURE 2: Comparison of convergence speeds of ratio $\%\mathrm{Vol}_{0.95}(\Omega)$ in the case $C_1 = C_2 = C_3 = 0.01$ and $C^{\mathrm{tot}} = 1$. m^{nest} : proposed nested approach. m^{AK} : AK-MCS approach. $m^{\mathrm{AK},Q^{(2)}}$: AK-MCS approach starting from the same computational cost than for the nested case.

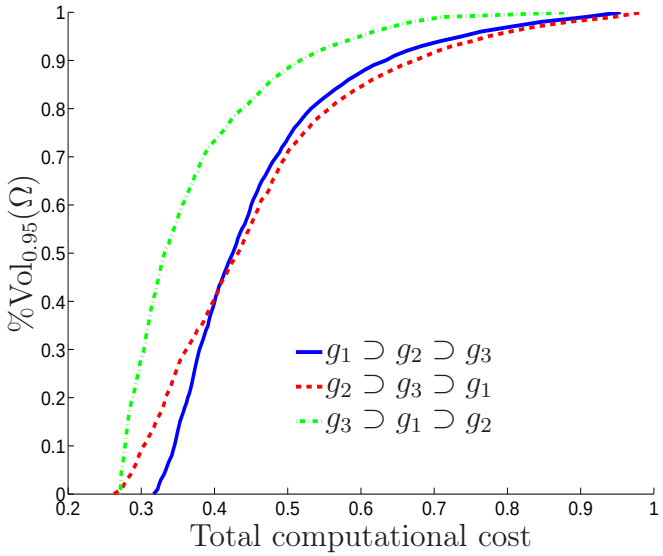

FIGURE 3: Influence of the classification of the performance functions on the convergence speed of ratio $\%\mathrm{Vol}_{0.95}(\Omega)$. For $1 \leq i, j, k \leq 3$, the symbol $g_i \supset g_j \supset g_k$ means that the surrogate models associated with g_i, g_j and g_k are respectively computed on $\mathbb{R}^d$, Ω_i and $\Omega_i \cap \Omega_j$.

equal to 0.01, and we fix to 1 the admissible total computational cost. Hence, at most 100 evaluations of g_1 g_2 or g_3 can be made to identify $\partial\Omega$.

A set of $Q = 16$ elements of $[0,1]^2$ (other values of Q would lead to the same conclusions), which are denoted by $\left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(16)}\right\}$, are then chosen randomly in $[0,1]^2$, to define the initial learning set for the evaluation of the GPR-based surrogate models associated with performance functions g_1, g_2 and g_3. Based on this learning set, the convergence of $\%\mathrm{Vol}_{0.95}(\Omega)$ with respect to the total computational cost is then evaluated. Such a procedure being repeated 100 times with 100 different initial learning sets, the mean values of $\%\mathrm{Vol}_{0.95}(\Omega)$, which are respectively written m^{AK} for the AK-MCS approach and m^{nest} for the proposed approach, are compared in Figure 2. Thus, it can be verified that limiting the initial number of evaluations by introducing nested surrogate models allows a quicker convergence of $\%\mathrm{Vol}_{0.95}(\Omega)$.

In order to reduce this influence of the initial computational cost for the AK-MCS approach, we denote by $Q_1^{(2)}, \ldots, Q_{100}^{(2)}$ the round value of the third of the initial computational cost associated with each convergence analysis for the proposed approach. For all $1 \leq j \leq 100$, the convergence of $\%\mathrm{Vol}_{0.95}(\Omega)$ for an AK-MCS approach starting from a $Q_j^{(2)}$-dimensional initial learning set is then computed. The evolution of the ratio $\%\mathrm{Vol}_{0.95}(\Omega)$ associated with these 100 repetitions is then added to Figure 2. The fact that, for the same computational cost, this ratio is always lower than the one associated with the proposed approach, emphasizes once again the interest of the proposed nested structure.

In this section, as the computational costs C_1, C_2 and C_3 are equal, performance functions g_1, g_2 and g_3 can be sorted in different ways. To quantify the influence of this initial sorting, Figure 3 compares the evolution of $\%\mathrm{Vol}_{0.95}(\Omega)$ with respect to the total computational cost for three particular sortings, where, for $1 \leq i, j, k \leq 3$, the symbol $g_i \supset g_j \supset g_k$ means that the surrogate models associated with g_i, g_j and g_k are respectively computed on $\mathbb{R}^d$, Ω_i and $\Omega_i \cap \Omega_j$. Such a figure therefore underlines the interest of sorting the performance functions, when it is possible, in order to minimize the domains on which the nested surrogate models are defined. In particular, it can be seen that the choice $g_3 \supset g_1 \supset g_2$ is very efficient for the proposed application.

4.2. Influence of the differences between the computational costs

When the computational costs associated with each performance functions are different, it is proposed in this work to sort them from the cheapest to the most expensive. To illustrate the influence of such a choice in terms of computational efficiency, three particular cases are studied :

— case 1 : $C_1 = 0.001$, $C_2 = 0.005$, $C_3 = 0.03$,

— case 2 : $C_1 = 0.005$, $C_2 = 0.03$, $C_3 = 0.001$,

— case 3 : $C_1 = 0.03$, $C_2 = 0.001$, $C_3 = 0.005$.

Moreover, two selection criteria, which are written SC_1 and SC_2, are introduced, such that :

$$\mathrm{SC}_1 : \quad (n^\star, \boldsymbol{x}^\star) = \arg \max_{1 \leq n \leq N, \quad \boldsymbol{x} \in \widetilde{\partial\Omega}} \widetilde{\sigma}_n(\boldsymbol{x}), \tag{22}$$

$$\mathrm{SC}_2 : \quad (n^\star, \boldsymbol{x}^\star) = \arg \max_{1 \leq n \leq N, \quad \boldsymbol{x} \in \widetilde{\partial\Omega}} \frac{\widetilde{\sigma}_n(\boldsymbol{x})}{C_n(\boldsymbol{x})}. \tag{23}$$

Therefore, in each case, we denote by $g_i \supset g_j \supset g_k - \mathrm{SC}_\ell$ the active learning procedure to identify Ω, which is based, first, on the nested surrogate models g_i, g_j and g_k that are respectively defined on $\mathbb{R}^d$, Ω_i and $\Omega_i \cap \Omega_j$, and second, on the selection criterion SC_ℓ. For each procedure and for each case, the evolutions of the mean value (over 100 repetitions of the same procedure) of $\%\mathrm{Vol}_{0.95}(\Omega)$ with respect to the total computational cost are compared in figures 4, 5 and 6. In these figures, it can be seen, first, that sorting the performance functions from the cheapest to the most expensive allows us to decrease the total computational cost for a given value of $\%\mathrm{Vol}_{0.95}(\Omega)$, and second, that integrating the computational cost in the selection criterion, which is the case for SC_2, leads to an additional convergence acceleration. At last, starting from particular 16-dimensional initial learning sets, figures 7, 8 and 9 represent the points where function g_1, g_2 and g_3 have been evaluated and used to update the different surrogate models to allow $\%\mathrm{Vol}_{0.95}(\Omega)$ to be greater than 90%, in the case of the optimal active learning procedure. For each case, a distinction is made between the points of the initial learning set, and the new points associated with each function g_j, which are denoted by $\left\{\boldsymbol{x}_i^{g_j},\ 1 \leq i \leq N_j\right\}$. It is reminded that, according to the algorithm defined by Eq. (1), the three performance functions are not evaluated in each point of the initial learning set. Hence, these figures show the ability of the proposed method to concentrate the performance functions evaluations on the boundaries of $\Omega = \Omega \cap \Omega_2 \cap \Omega_3$, and to limit the calls to the most expensive

Computational costs	N_1	N_2	N_3
$C_1 = 0.001$, $C_2 = 0.005$, $C_3 = 0.03$	25	14	25
$C_1 = 0.005$, $C_2 = 0.03$, $C_3 = 0.001$	23	12	43
$C_1 = 0.03$, $C_2 = 0.001$, $C_3 = 0.005$	14	21	19

TABLE 1: Influence of the computational costs $(C_n)_{1\leq n\leq 3}$ of each performance functions $(g_n)_{1\leq n\leq 3}$ on the minimal number of evaluations $(N_n)_{1\leq n\leq 3}$ needed for $\%\mathrm{Vol}_{0.95}(\Omega)$ to be greater than 90%.

performance functions. Indeed, as it can be seen in Table 1, which shows the number of calls to each functions for these three particular cases, for the same precision on $\%\mathrm{Vol}_{0.95}(\Omega)$, the total number of evaluations of functions g_1, g_2 and g_3 can be very different, with respect to its computational cost.

5. Conclusions

For the last decade, the use of a surrogate model for the conception and the reliability evaluation of complex systems has kept increasing. Indeed, it is a very powerful tool to reduce the computational costs associated with the estimation of the limits of a system at a given precision. When these limits are characterized by performance functions, such surrogate models are generally based on the introduction of selection criteria to iteratively choose the new points to be evaluated to improve the classification precision of the definition domain. In that prospect, this paper first proposed to sort the performance functions with respect to their computational cost, and to introduce nested GPR-surrogate models to avoid as much as possible useless evaluations of the performance functions. Then, an original selection criterion was defined, which takes into account the computational costs associated with each performance function, in order to maximize the knowledge about the limit states of the system at the minimal computational cost. At last, the potential of such an active learning algorithm for the conception and the reliability analyses of complex systems has been illustrated on an analytical example.

In the proposed procedure, it has to be noticed that all the performance functions depend on the same inputs. Dealing with performance functions sharing a limited number of inputs is an interesting perspective for future work. In the same manner, only fixed computational costs have been considered in the present work. In practice, these costs can depend on the values

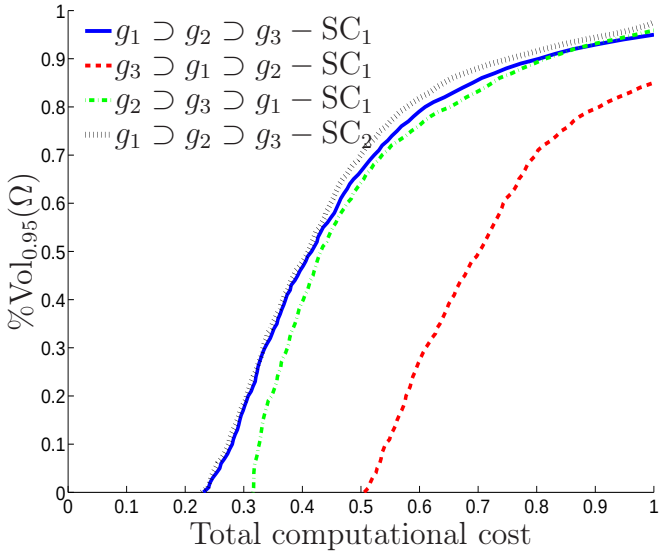

FIGURE 4: Influence of the classification of the performance functions and of the selection criterion SC_1 or SC_2 on the convergence speed of ratio $\%\mathrm{Vol}_{0.95}(\Omega)$ when $C_1 = 0.001$, $C_2 = 0.005$, $C_3 = 0.03$.

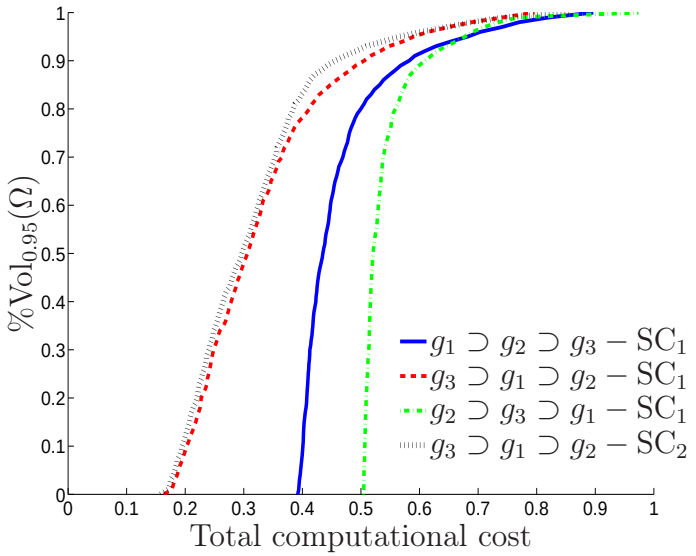

FIGURE 5: Influence of the classification of the performance functions and of the selection criterion SC_1 or SC_2 on the convergence speed of ratio $\%\mathrm{Vol}_{0.95}(\Omega)$ when $C_1 = 0.005$, $C_2 = 0.03$, $C_3 = 0.001$.

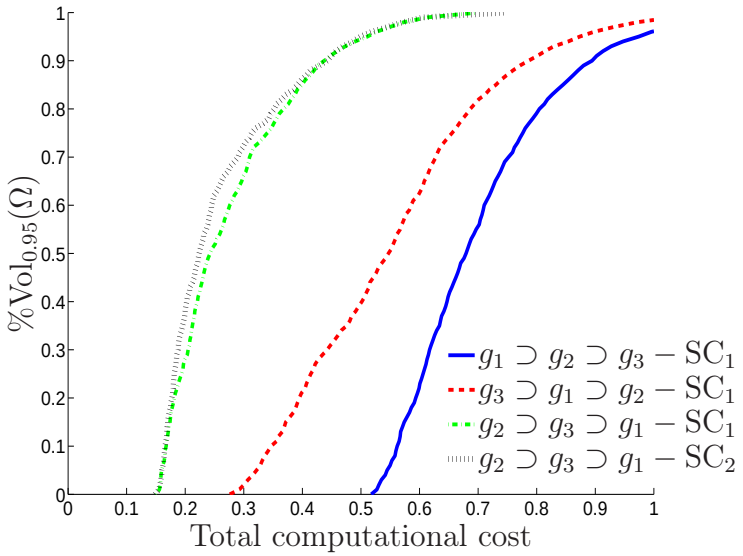

FIGURE 6: Influence of the classification of the performance functions and of the selection criterion SC_1 or SC_2 on the convergence speed of ratio $\%\mathrm{Vol}_{0.95}(\Omega)$ when $C_1 = 0.03$, $C_2 = 0.001$, $C_3 = 0.005$.

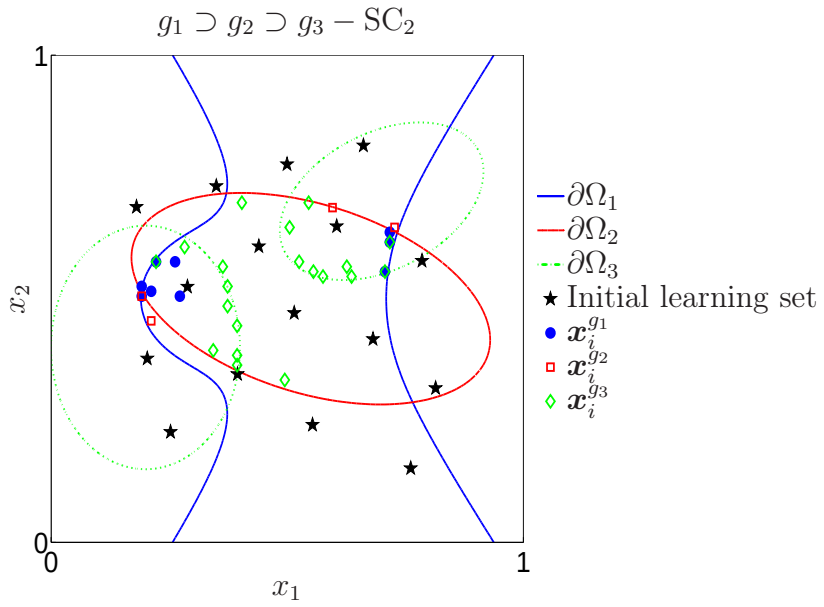

FIGURE 7: Positions of the points where functions g_1, g_2 and g_3 have been evaluated when $C_1 = 0.001$, $C_2 = 0.005$, $C_3 = 0.03$.

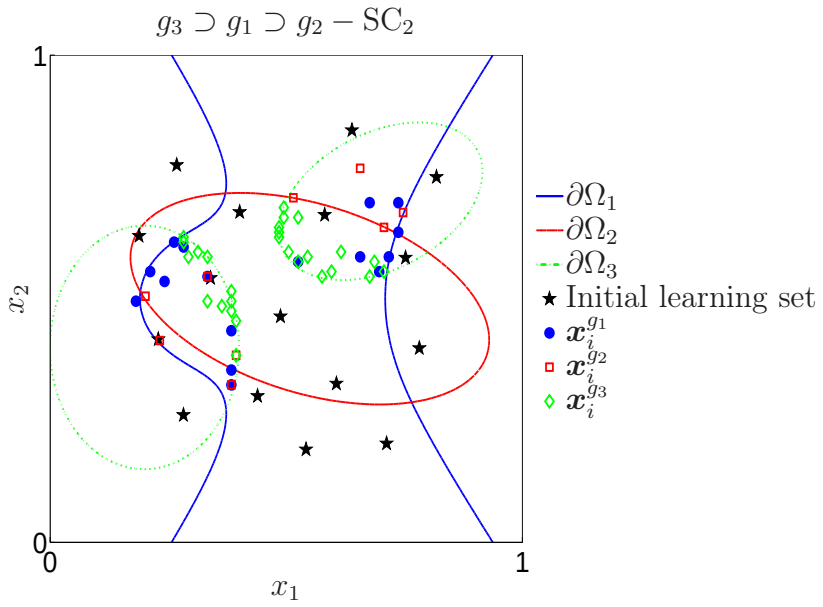

FIGURE 8: Positions of the points where functions g_1, g_2 and g_3 have been evaluated when $C_1 = 0.005$, $C_2 = 0.03$, $C_3 = 0.001$.

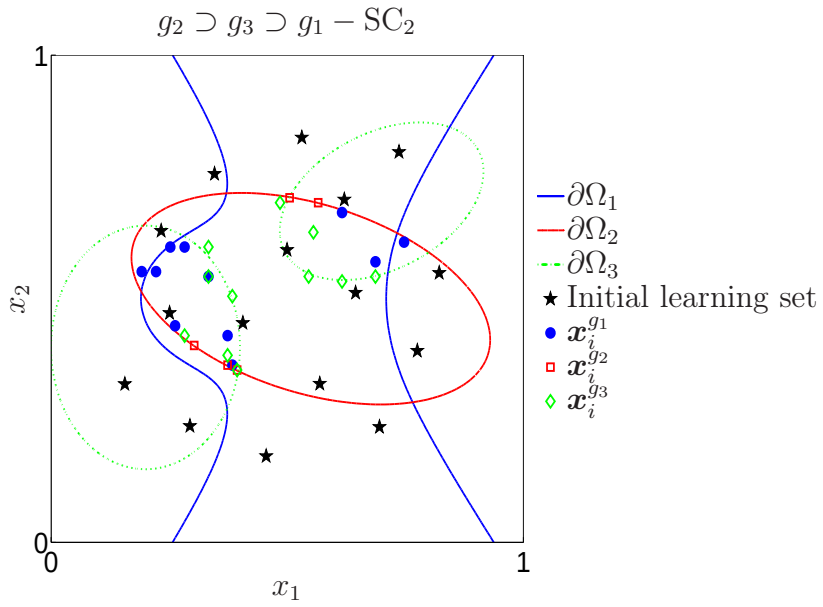

FIGURE 9: Positions of the points where functions g_1, g_2 and g_3 have been evaluated when $C_1 = 0.03$, $C_2 = 0.001$, $C_3 = 0.005$.

of the inputs, as a configuration can be more complicated to simulate than an other, especially when considering time-dependent non linear equations. Proposing methods that both predict the costs associated with the evaluations of the performance functions and take them into account in the selection criterion is a possible extension of the present work. In addition, the adaptation of the presented criterion, which takes into account the computational cost in the sampling strategy, for methods based on global measures of uncertainty, such as SUR, is also an interesting perspective that could provide alternative sampling criteria for the identification of the boundary of a conception domain.

At last, the proposed approach being based on the Gaussian process regression, it is affected by the same difficulties when confronted to high dimensional problems. Enabling the GPR to deal with systems with high values for d is still a very important challenge for many industrial applications.

[1] J Bect, D Ginsbourger, L Li, V Picheny, and E Vasquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22 :773–793, 2012.

[2] B Bichon, J McFarlan, and S Mahadevan. Efficient surrogate models for reliability %analysis of systems with multiple failure modes. *Reliab. Eng. Syst. Safety*, 96 :1386–95, 2011.

[3] B J Bichon, M S Eldred, L P Swiler, S Mahadevan, and J M McFarland. Efficient global reliability analysis for non linear implicit performance functions. *AIAA Journal*, 46(10) :2459–2468, 2008.

[4] E Castillo, J M Sarabia, C Solares, and P Gomez. Uncertainty analyses in fault trees and Bayesian networks using FORM/SORM methods. *Reliability Engineering and System Safety*, 1999.

[5] C Chevalier, J Bect, D Ginsbourger, E Vazquez, V Picheny, and Y Richet. Fast kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics, Taylor & Francis*, 56(4) :455–465, 2014.

[6] B Echard, N Gayton, and M Lemaire. AK-MCS : an active learning reliability method combining Kriging and Monte Carlo simulation. *Struct. Safety*, 33 :145–154, 2011.

[7] K T Fang, R Li, and A Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall, Computer Science and Data Analysis Series, London, 2006.

[8] W Fauriat and N Gayton. AK-SYS : An adaptation of the AK-MCS method for system reliability. *Reliability Engineering and System Safety*, 123 :137–144, 2014.

[9] A N Freudenthal. Safety and the probability of structural failure. *Transactions ASCE*, 121 :1337–1397, 1956.

[10] F Hayashi. *Econometrics*. Princeton University Press, 2000.

[11] J Hortado. *Structural reliability - Statistical learning perspectives*, volume 17. Lectures notes in applied and computational mechanics, Springer, 2002.

[12] M C Kennedy and A O'Hagan. Bayesian calibration of computer models. *Journal of the royal statistical society*, 63 :425–464, 2001.

[13] M Lemaire. *Structural reliability*. ISTE/Wiley, 2009.

[14] R Rackwitz and B Fiessler. Structural reliability under combined load sequences. *Journal of computer structures*, 9 :479–484, 1978.

[15] J Sacks, W Welch, T Mitchell, and H Wynn. Design and analysis of computer experiments. *Statistical Science*, 4 :409–435, 1989.

[16] T J Santner, B J Williams, and W I Notz. *The design and analysis of computer experiments*. Springer, New York, 2003.

[17] B Scholkopf and A J Smola. *Learning With Kernels : Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Boston, 2002.

[18] A Takeshi. *Advanced econometrics*. Harvard University Press, 1985.

[19] V Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.