Robustness evaluation of the reliability of penstocks combining line sampling and neural networks

Antoine Ajenjo, Emmanuel Ardillon, Vincent Chabridon, Scott Cogan, Emeline Sadoulet-Reboul

Robustness evaluation of the reliability of penstocks combining line sampling and neural networks

Antoine Ajenjo[a,b,c], Emmanuel Ardillon[a], Vincent Chabridon[a], Scott Cogan[b], Emeline Sadoulet-Reboul[b]

[a] *EDF R&D, 6 quai Watier, 78401 Chatou, France*
[b] *Univ. Bourgogne Franche-Comté, CNRS/UFC/ENSMM/UTBM, Department of Applied Mechanics, 24 rue de l'épitaphe, 25000 Besançon, France*
[c] *Corresponding Author*

Abstract

The objective of this work is to conduct robustness evaluations on the reliability assessment of penstocks using the info-gap framework. In order to improve the induced optimization searches, three original line sampling procedures are proposed in order to address the complex limit-state function on which the failure probability depends. The proposed algorithms are proven to be well suited for the search of the multiple roots involved in the line sampling technique. Then, a classification and a regression artificial neural network are combined for predicting the roots in order to reduce the computational time engendered by robustness evaluations.

Keywords: structural reliability, line sampling, info-gap, neural networks, robustness

1. Introduction

Structural reliability [1] is of particular interest for risk-sensitive industrial applications such as power generation [2] where system performance, and therefore safety, is subject to uncertainty. In this context, the safety is assessed by estimating reliability-oriented quantities of interest such as a low failure probability or a high-order quantile on a specific output variable of interest. Two types of uncertainty are commonly distinguished, namely aleatory and epistemic [3]. Aleatory uncertainty is associated with natural randomness while epistemic uncertainty is understood as ignorance due to a lack of knowledge and is therefore potentially reducible. High-risk systems

models are typical cases where epistemic uncertainty can be encountered as they often represent events that are rarely or never realized. However, the potential impact of lack of knowledge must still be accounted for in order to improve information for a more reliable decision-making process regarding the safety of the system.

The notion of robustness has many interpretations and possible mathematical representations [4]. In the present paper, it is defined as the capacity of the system to fulfill a criterion despite differences between its predicted and operational behaviors which is a key point in engineering and more specifically in safety assessment. The info-gap framework [5] proposes a metric that quantifies the robustness of a possible decision with respect to (w.r.t.) epistemic uncertainty by calculating its worst performance at increasing levels of uncertainty in order to privilege tolerance to unexpected situations over performance at a poor estimate of the system's environment [6]. Info-gap has been applied in a wide range of fields where decisions under severe uncertainty need to be made such as in structural design [7], climate policies [8] or water resource planning [9]. In [10], the probabilistic framework and the info-gap framework are combined considering uncertainty on a covariance matrix. However, its application to reliability quantities of interest such as failure probabilities has been less studied although an example can be found in [11] in the context of hybrid reliability analysis. Yet, the info-gap framework is particularly relevant in the context of rare event analysis [12] in which this work falls.

In this paper, the info-gap method is applied to a real world industrial reliability model assessing the mechanical integrity of penstocks by evaluating a failure probability. As the uncertain parameters involve probabilistic distribution parameters, assessing the info-gap robustness of the model reduces to evaluating maximum failure probabilities for a series of increasing parametric probability boxes problems [13]. This requires an efficient failure probability estimator, both in terms of global precision over the uncertainty space and computational time. The former requirement is challenging when assessing the reliability of penstocks as the failure event corresponds to a restricted intersection domain of complex geometry. A wide range of approximation and sampling methods are available for estimating failure probabilities [14] and some of them are already used for assessing the reliability of penstocks. In the present paper, the technique known as line sampling [15] is applied to better target the intersection domain. Three adapted line sampling algorithms considering three equivalent formulations of the intersection failure

event are proposed in order to efficiently evaluate the associated roots which constitutes the main challenge induced by this technique. The performances of these algorithms are analyzed through info-gap robustness curves. Secondly, a methodology based on two deep neural networks is considered in order to predict the roots involved in the line sampling algorithms considering the aleatory and epistemic spaces jointly. This enables to considerably reduce the computational burden that may be caused by an info-gap analysis.

This work shows how customized line sampling algorithms may be competitive even for reliability problems with complex limit-state functions. Moreover, it provides an example on how neural networks can be used to help assess failure probabilities based on line sampling as soon as the computer model is affordable to run.

The paper is organized as follows: Section 2 reviews the formulation of a reliability analysis and presents the specific case of the reliability of penstocks; Section 3 presents the three proposed line sampling algorithms; Section 4 describes the methodology applied for building robustness curves and validates the line sampling algorithms; finally Section 5 proposes a combination of two artificial neural networks with the line sampling algorithm to reduce computational time. Conclusions and perspectives are drawn in Section 7.

2. Reliability assessment of penstocks for hydroelectric facilities

2.1. General formulation of a reliability problem

The objective of a reliability analysis is to assess the safety of a system subject to uncertainty. The safety is evaluated through the limit-state function $g(\mathbf{x})$ defined such that the event $g(\mathbf{x}) \leq 0$ represents a failure state of the system. Hence the failure domain is given by $\mathcal{F} = \{\mathbf{x} \in D_{\mathbf{X}}, g(\mathbf{x}) \leq 0\}$. Probability theory offers a framework to propagate aleatory uncertainty on the input vector through the model. The input vector of uncertain variables is modeled as a random vector $\mathbf{X}$ to which a supposedly known joint probability density function (pdf) $f_{\mathbf{X}}$ is attributed. After propagating the uncertainty through the limit-state function, the output $Z = g(\mathbf{X})$ is also a random variable. The exact pdf f_Z is generally inaccessible but reliability quantities of interest can be estimated such as moments or quantiles. In this work, the failure probability P_{f} is of interest:

$$P_{\mathrm{f}} = \Pr\left[g(\mathbf{X}) \leq 0\right] = \int_{\mathcal{F}} f_{\mathbf{X}}(\mathbf{x})\, d\mathbf{x}. \tag{1}$$

Generally in complex systems, techniques are needed to evaluate Eq. (1) such as sampling methods (Monte Carlo, importance sampling) or approximation methods (first/second-order reliability method) [14]. In this work, the line sampling (LS) method [15, 16] is investigated to better reach a geometrical complex limit-state function. This algorithm will be presented in details in Section 3.

2.2. Presentation of the penstock model

This work focuses on an industrial use-case relevant to the French electricity company EDF which concerns the reliability study of penstocks [17]. EDF operates more than 500 penstocks having a total length of over 300 km. Penstocks are pipes made of steel used to transport water under pressure from the water dam to the hydroelectric turbine. Due to thickness loss resulting from corrosion, their mechanical integrity must be justified. The usual justification relies on diagnoses involving thickness measurements and the evaluation of a deterministic margin factor (MF) which is a ratio of an allowable mechanical stress over the current mechanical stress present in the pipe during operation. If this ratio is greater than one, then the penstock is considered as fit for service. The evaluation of MF depends on many variables which mainly pertain to mechanical and geometrical properties. The integrity needs to be justified for a very large panel of penstocks with different properties (e.g., geometry, mechanical properties) which justifies the use of a predictive mechanical model. Uncertainty on some variables may affect a deterministic evaluation of this model. A historical conservative approach consists in evaluating the MF when attributing penalized values on the uncertain variables. The next section presents another approach which treats uncertainty with probabilistic distributions.

2.3. Reliability model of penstocks

To optimize the MF, a general reliability approach was developed to assess the failure probability at year $N + 1$ of a given penstock. Two major failure modes have been identified and investigated: plastic collapse (affecting parent metal) and brittle failure (affecting welds), due to the presence of cracks appearing during the welding process. In the present application, only the second failure mode is considered since its reliability analysis is the most complex one:

- the limit-state function is locally non-differentiable and can be discontinuous;

- the annual failure probability estimated here is a conditional probability considering that the penstock passed a hydraulic pressure test (HPT) after its production in the workshop.

For the sake of simplicity, the dependence w.r.t. $\mathbf{X}$ is omitted in $G_i = G_i(\mathbf{X})$. The conditional failure probability at year $N+1$ can be expressed as:

$$P_{\mathrm{f}} = \Pr\left(G_{N+1} < 0 \cap G_N \geq 0 \mid G_{\mathrm{HPT}} \geq 0\right) \tag{2}$$

which leads to (using Bayes theorem):

$$P_{\mathrm{f}} = \frac{\Pr\left(G_{N+1} < 0 \cap G_N \geq 0 \cap G_{\mathrm{HPT}} \geq 0\right)}{\Pr\left(G_{\mathrm{HPT}} \geq 0\right)} \tag{3}$$

where G_{N+1} is the limit-state function at the beginning of year $N+1$, G_N is the limit-state function at the beginning of year N and $\{G_{\mathrm{HPT}} \geq 0\}$ is the event meaning that the penstock successfully passed the hydraulic pressure test. In the following, only the numerator in Eq. (3) is of interest as it is the most challenging one to estimate. As the G-functions decrease over time due to the monotonic corrosion degradation, the following expression holds:

$$\Pr\left(G_{N+1} < 0 \cap G_N \geq 0\right) = \Pr\left(G_{N+1}.G_N < 0\right). \tag{4}$$

The numerator in Eq. (3) corresponds to the probability of an intersection of three events which is depicted as the red hatched band in Figure 1. This probability is usually very small (e.g., smaller than 10^{-6}). In the following, the double intersection will be handled with three equivalent events:

$$\begin{aligned} E_1 &= \{\max\left(G_{N+1}, -G_N, -G_{\mathrm{HPT}}\right) \leq 0\}, \\ E_2 &= \{G_{N+1}.G_N \leq 0 \cap G_{\mathrm{HPT}} > 0\}, \\ E_3 &= \{G_{N+1} \leq 0 \cap G_N > 0 \cap G_{\mathrm{HPT}} > 0\}. \end{aligned} \tag{5}$$

Although being equivalent (due to a monotonic decreasing behavior of the limit-state functions w.r.t. time), these different formulations will induce different strategies on the failure probability estimation as it will be seen in Subsection 3.3 for the root search involved with line sampling. The following notations are used: $G_{\max} = \max\left(G_{N+1}, -G_N, -G_{\mathrm{HPT}}\right)$ and $G_{\mathrm{prod}} = G_{N+1}.G_N$. The expressions of G_{N+1}, G_N and G_{HPT} depend on quite a lot of parameters and functions describing the mechanical behavior of the penstock under hydraulic pressure. The reader is referred to [18] for a more detailed overview

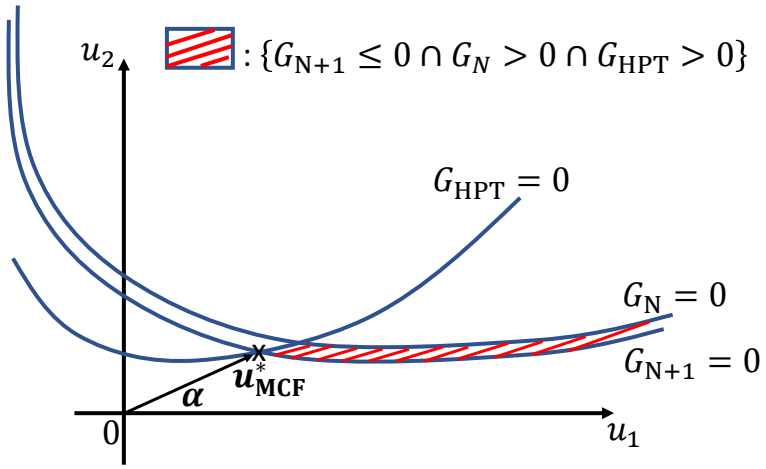


Figure 1: Illustration of a possible failure domain and the basic principle of the multi-constraint FORM (MCF) algorithm used in the FISTARR algorithm.

of the mechanical model. However, the most important feature about these functions is that they remain analytical which allows them to be computed rapidly.

The probabilistic vector $\mathbf{X}$ is of dimension $n_{\mathbf{X}} = 6$ and its components are detailed in Table 1 where R_m is the ultimate tensile strength, ε a parameter used to linearly express the yield strength as a function of R_m, Δe_{corr} the thinning due to water and atmospheric corrosion, Δe_{extra} the extra thickness added to the design thickness, a is the height of the crack and K_{IC} the tenacity of the material.

Table 1: Input probabilistic modeling of $\mathbf{X}$ for the penstock use-case.

X_i	Distribution	param. 1	param. 2	param. 3
$X_1 = R_m$ (MPa)	Lognormal	μ_{R_m}	σ_{R_m}	-
$X_2 = \varepsilon$ (MPa)	Normal	μ_{ε}	σ_{ε}	-
$X_3 = \Delta e_{\text{corr}}$ (mm)	Normal	$\mu_{\Delta e_{\text{corr}}}$	$\sigma_{\Delta e_{\text{corr}}}$	-
$X_4 = \Delta e_{\text{extra}}$ (mm)	Normal	$\mu_{\Delta e_{\text{extra}}}$	$\sigma_{\Delta e_{\text{extra}}}$	-
$X_5 = a$ (mm)	Uniform	0	$a_{\max}$	-
$X_6 = K_{\text{IC}}$ (MPa.$\sqrt{\text{m}}$)	Weibull Min	$\beta_{K_{\text{IC}}}$	$\alpha_{K_{\text{IC}}}$	$\gamma_{K_{\text{IC}}}$

Standard penstock reliability assessments have been performed for some

years and, unlike info-gap robustness evaluations of reliability assessments, consider large variations of parameters to reproduce the variety of operating penstocks. They resort to a panel of reliability assessment methods capitalized in the OpenTURNS library-based [19] Persalys-Penstock software [18]. The most efficient method in the context of standard penstock reliability assessments is the so-called FISTARR method (for **F**ORM-**IS**-**T**ested **A**utomatically-**R**apid sea**R**ch [18]), an extended adaptation of FORM-IS for multiple intersection events. More precisely this adaptation consists in a selection of FORM-IS algorithms including MCF-IS algorithms, an importance sampling around the design point obtained by performing a multi-constraint-FORM (MCF) analysis as shown in Figure 1.

In the present paper, only the MCF-IS algorithm from FISTARR, applied with the LD_MMA optimization algorithm from the NLopt Python Library, is used and will be simply referred as "IS" in order to simplify the notation. Despite this method being globally robust in its current version, that could be further optimized, it may not always converge rapidly for a few configurations of input probabilistic parameters. In the following section, a LS procedure based on the MCF design point is presented in order to investigate a new technique for the estimation of the failure probability of penstocks in the context of info-gap robustness assessments of penstock reliability assessments.

3. A new line-sampling-based procedure adapted to multiple roots

3.1. Generalities on line sampling

This algorithm, also known as "Axis-Orthogonal Simulation" [20], consists in, firstly, generating samples in a hyperplane orthogonal to a direction $\boldsymbol{\alpha}$ that points towards the limit-state surface, and then, solving several line searches in that direction. A first preliminary step is to apply an isoprobabilistic transformation $T : \mathbb{R}^{n_{\mathbf{X}}} \longrightarrow \mathbb{R}^{n_{\mathbf{X}}}$ which maps the original random variables $\mathbf{X}$ with the standard normal random variables $\mathbf{U}$ where $n_{\mathbf{X}}$ is the number of random variables. By applying the rotation $\mathbf{R}$ such that $\mathbf{V} = \mathbf{R}\mathbf{U}$, where V_1 is a standard random variable whose outcome is parallel to $\boldsymbol{\alpha}$ and $\mathbf{V}_{\mathbf{2:n_X}} = \mathbf{U}^{\perp}_{\boldsymbol{\alpha}}$ is a random vector (of size $(n_{\mathbf{X}} - 1)$) whose realization lies in the hyperplane orthogonal to $\boldsymbol{\alpha}$, the failure probability can be expressed as follows:

$$P_{\mathrm{f}} = \int_{\mathbb{R}^{n_{\mathbf{X}}-1}} \int_{G^{\perp} \leq 0} \varphi\left(v_1\right) dv_1 \varphi_{\mathbf{U}^{\perp}_{\boldsymbol{\alpha}}}\left(\mathbf{u}^{\perp}_{\boldsymbol{\alpha}}\right) d\mathbf{u}^{\perp}_{\boldsymbol{\alpha}} \tag{6}$$

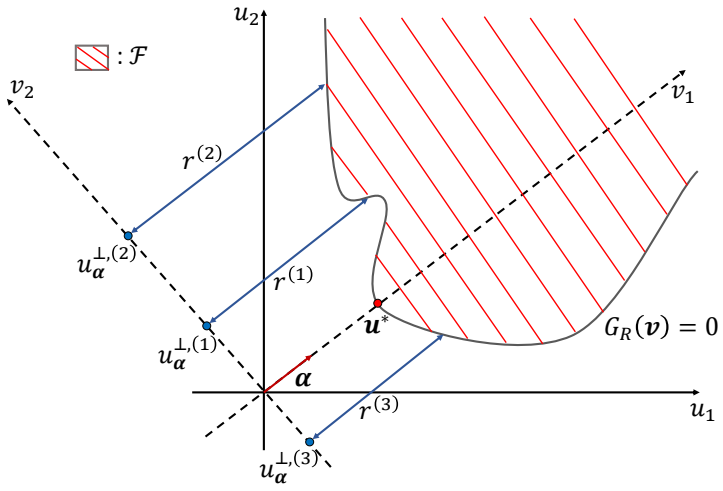


Figure 2: Graphical representation of the LS procedure.

where $G^{\perp}\left(\left[v_1, \mathbf{u}_{\boldsymbol{\alpha}}^{\perp}\right]\right)$ is the limit-state function in the rotated space and $\varphi(\cdot)$ is the pdf of the standard normal distribution. A graphical representation of the LS procedure is given in Figure 2. Assuming that, for any $\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}$, $r\left(\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}\right)$ is the unique solution of $G^{\perp}\left(v_1; \mathbf{u}_{\boldsymbol{\alpha}}^{\perp}\right) = 0$, the failure probability can finally be expressed as:

$$P_{\mathrm{f}} = \int_{\mathbb{R}^{n\mathbf{x}-1}} \Phi\left(-r\left(\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}\right)\right) \varphi_{\mathbf{U}_{\boldsymbol{\alpha}}^{\perp}}(\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}) d\mathbf{u}_{\boldsymbol{\alpha}}^{\perp} \tag{7}$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution. When sampling n_{LS} orthogonal points, the estimation of the failure probability and its corresponding variance can be estimated as follows:

$$\widehat{P}_{\mathrm{f}} = \frac{1}{n_{\mathrm{LS}}} \sum_{i=1}^{n_{\mathrm{LS}}} p_{\mathrm{f}}^{(i)} \tag{8}$$

$$\mathrm{Var}\left(\widehat{P}_{\mathrm{f}}\right) = \frac{1}{n_{\mathrm{LS}}\left(n_{\mathrm{LS}} - 1\right)} \sum_{i=1}^{n_{\mathrm{LS}}} \left(p_{\mathrm{f}}^{(i)} - \widehat{P}_{\mathrm{f}}\right)^2 \tag{9}$$

where $p_{\mathrm{f}}^{(i)} = \Phi\left(-r\left(\mathbf{u}_{\boldsymbol{\alpha}}^{\perp,(i)}\right)\right)$ is the failure probability along the $(i)^{\mathrm{th}}$-line.

3.2. Challenge for applying line sampling to the penstock reliability problem

As the limit-state function defined in Subsection 2.3 for the reliability model for penstocks represents the intersection of three events, an adaptation

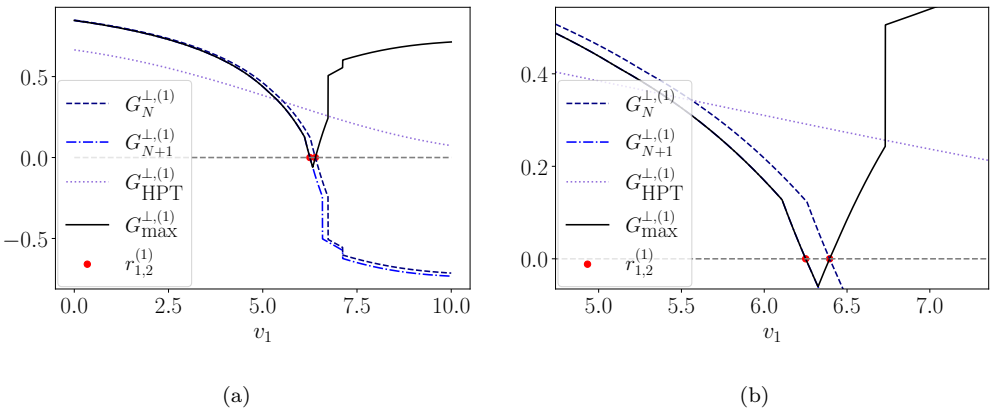
of the LS procedure presented in Subsection 3.1 is required. Indeed, the assumption of unicity of the root $r^*(\mathbf{v_{2:n}})$ is not met here. Applications of LS to such cases can already be found such as in [21] in the context of linear limit-state functions and Gaussian random variables.

In the particular case of the reliability of penstocks, Figure 1 illustrates the fact that, depending on the sample $\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}$ and the direction $\boldsymbol{\alpha}$, two cases can be encountered:

- either the failure band is not reached, which results in the fact that there is no root;
- or the failure band is reached in which case there are two roots denoted r_1 and r_2.

Figures 3 and 4 represent the several limit-state functions $G_{N+1}^{\perp}(v_1)$, $G_N^{\perp}(v_1)$, $G_{\text{HPT}}^{\perp}(v_1)$ and $G_{\max}^{\perp}(v_1)$ for two different samples $\mathbf{u}_{\boldsymbol{\alpha}}^{\perp,(i)}$ where $G_j^{\perp} = G_j\left(\mathbf{u}_{\boldsymbol{\alpha}}^{\perp,(i)}, v_1\right)$.

As the functions $G_{N+1}^{\perp}$, $G_N^{\perp}$ and $G_{\text{HPT}}^{\perp}$ are decreasing, the composed func-



(a) (b)

Figure 3: Values of $G_j^{\perp,(i)}$ in function of v_1 for a case with two roots (a) on which a zoom is performed (b).

tion $G_{\max}^{\perp}$ first decreases with $G_{N+1}^{\perp}$ and then increases either with $-G_N^{\perp}$ or $-G_{\text{HPT}}^{\perp}$. Actually, there are three distinguishable cases:

1. the two roots correspond to $G_{N+1}^{\perp} = 0$ and $G_N^{\perp} = 0$ as in the example in Figure 3;
2. the root of $G_{\text{HPT}}^{\perp} = 0$ is smaller than the root of $G_{N+1}^{\perp} = 0$ which implies no solution for $G_{\max}^{\perp} = 0$ as in the example in Figure 4,

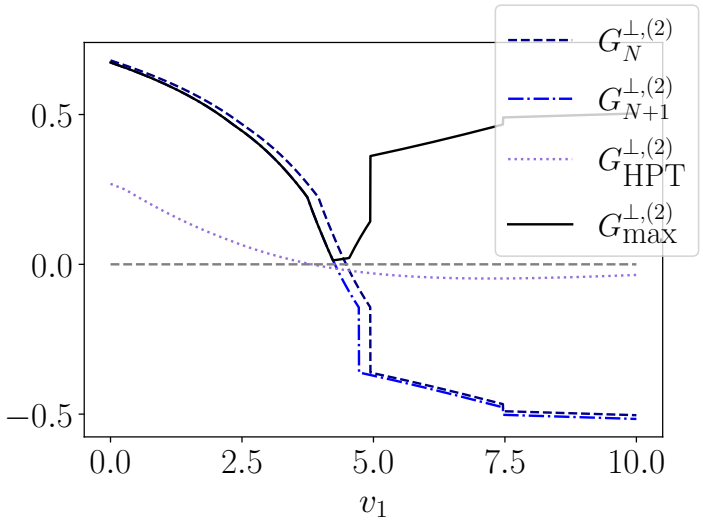


Figure 4: Values of $G_j^{\perp,(i)}$ in function of v_1 for a case with no roots (see the $G_{\max}^{\perp,(2)}$ curve).

3. the two roots correspond to $G_{N+1}^{\perp}$ and $G_{\text{HPT}}^{\perp}$ (much rarer). Typically, this happens when the root of $G_{\text{HPT}}^{\perp} = 0$ is between the roots of $G_{N+1}^{\perp} = 0$ and $G_N^{\perp} = 0$.

When performing LS with two roots r_1 and r_2 (such that $r_1 < r_2$) involved at each iteration, $p_{\text{f}}^{(i)}$ from Eq. (8) becomes:

$$p_{\text{f}}^{(i)} = \Phi\left(-r_1^{(i)}\right) - \Phi\left(-r_2^{(i)}\right). \tag{10}$$

In such case where no roots exist, $p_{\text{f}}^{(i)}$ is equal to 0. In the following section, three algorithms constructed in accordance with the events E_1, E_2 and E_3 are proposed in order to efficiently solve this multiple root search problem.

3.3. Proposition of three adapted line sampling procedures

Algorithm 1 presents the general LS procedure (in the standard normal space) used to estimate the failure probability which is very common, except for the fact that the initial direction $\boldsymbol{\alpha}$ is obtained with a MCF algorithm and that there are two roots to search for (which may not always exist). If valid roots are found and if the point corresponding to the first root has a smaller distance to the origin than the previous optimal point $\mathbf{u}^*$, then

Algorithm 1 – General LS procedure (for possible two roots).

$P_\mathrm{f} \leftarrow 0$
Find $\boldsymbol{\alpha}, \mathbf{u}^*_\mathrm{MCF}$ # MCF results
Generate $\mathbf{u} \sim \mathcal{N}(0,1)$
for $i : 1 \rightarrow n_\mathrm{LS}$ **do**
  ${\mathbf{u}_\alpha}^{\perp,(i)} \leftarrow \mathbf{u}^{(i)} - \left(\mathbf{u}^{(i)}.\boldsymbol{\alpha}\right)\boldsymbol{\alpha}$ # projection on the orthogonal hyperplane
  Search for $r_1^{(i)}$ and $r_2^{(i)}$ # see Algorithms 2 and 3
  if $r_1^{(i)}$ and $r_2^{(i)}$ exist **then**
    $p_\mathrm{f}^{(i)} = \Phi\left(-r_1^{(i)}\right) - \Phi\left(-r_2^{(i)}\right)$ # failure lies in $\left[r_1^{(i)}, r_2^{(i)}\right]$
    if $\|{\mathbf{u}_\alpha}^{\perp,(i)} + r_1^{(i)}\boldsymbol{\alpha}\| < \|\mathbf{u}^*_\mathrm{MCF}\|$ **then** # active line sampling
      $\mathbf{u}^* \leftarrow {\mathbf{u}_\alpha}^{\perp,(i)} + r_1^{(i)}\boldsymbol{\alpha}$
      $\boldsymbol{\alpha} \leftarrow \mathbf{u}^*/\|\mathbf{u}^*\|$
    end if
  else
    $p_\mathrm{f}^{(i)} = 0$ # failure is never reached
  end if
  $P_\mathrm{f} \leftarrow P_\mathrm{f} + p_\mathrm{f}^{(i)}$
end for
$P_\mathrm{f} \leftarrow P_\mathrm{f}/n_\mathrm{LS}$

$\mathbf{u}^*$ and the optimal direction $\boldsymbol{\alpha}$ are updated. This is a feature of the so-called "active line sampling" [22] which is useful in this case as the MCF algorithm may not always give the best possible direction. The procedures used to find the roots when considering the events E_1 (related to G_max) and E_2 (related to G_prod) are both presented in Algorithm 2 as they are quite similar. Algorithm 3 is proposed to find the roots when considering only the event E_3. It is important to keep in mind that the choice of the event does not impact the position of the roots but only the procedure to find them. Indeed, the event E_1 is only composed of one function which is supposed to be always decreasing first and then always increasing. As it is formulated as the maximum value of three different functions, its shape may not be smooth. The corresponding procedure in Algorithm 2 (in blue) aims at estimating the minimum value of the function. If the minimum value is negative, then the first root and the second root are searched in its neighborhood (before and after). If the minimum value is positive, then no root exists and p_f is set to

Algorithm 2 – Roots search with the events E_1 and E_2.

$m^{(i)} = \min_{v_1} G^{\perp}_{\max}(v_1)$
$m^{(i)} = \min_{v_1} G^{\perp}_{\text{prod}}(v_1)$
if $m^{(i)} < 0$ **then** # else case 2
  Find $r_1^{(i)} < m^{(i)}$ s.t. $G^{\perp}_{\max}\left(r_1^{(i)}\right) = 0$
  Find $r_2^{(i)} > m^{(i)}$ s.t. $G^{\perp}_{\max}\left(r_2^{(i)}\right) = 0$ # case 1 or 3
  Find $r_1^{(i)} < m^{(i)}$ s.t. $G^{\perp}_{\text{prod}}\left(r_1^{(i)}\right) = 0$
  if $G^{\perp}_{\text{HPT}}\left(r_1^{(i)}\right) > 0$ **then** # else case 2
    Find $r_2^{(i)} > m^{(i)}$ s.t. $G^{\perp}_{\text{prod}}\left(r_2^{(i)}\right) = 0$ # case 1
    if $G^{\perp}_{\text{HPT}}\left(r_2^{(i)}\right) < 0$ **then**
      Find $r_2^{(i)}$ s.t. $G^{\perp}_{\text{HPT}}\left(r_2^{(i)}\right) = 0$ # case 3
    end if
  end if
end if

Algorithm 3 – Roots search with the event E_3.

Find $r_1^{(i)}$ s.t. $G^{\perp}_{N+1}\left(r_1^{(i)}\right) = 0$
if $G^{\perp}_{\text{HPT}}\left(r_1^{(i)}\right) > 0$ **then** # else case 2
  Find $r_2^{(i)} > r_1^{(i)}$ s.t. $G^{\perp}_{N}\left(r_2^{(i)}\right) = 0$ # case 1
  if $G^{\perp}_{\text{HPT}}\left(r_2^{(i)}\right) < 0$ **then**
    Find $r_2^{(i)}$ s.t. $G^{\perp}_{\text{HPT}}\left(r_2^{(i)}\right) = 0$ # case 3
  end if
end if

0. The procedure applied for the event E_2 and described in Algorithm 2 (in red) is quite similar to the one for the event E_1. E_2 is composed of the two functions G_{HPT} which is supposed always decreasing and $G_{\text{prod}} = G_{N+1}.G_N$ which is supposed always decreasing first and then always increasing. It is necessary, in this case, to verify the position of the root of G_{HPT}. If it appears

before the first root of G_{prod} then there is no solution. Otherwise, the second root to be kept is the smallest one between the root of G_{HPT} and the second root of G_{prod}.

The event E_3 is composed of the three supposed decreasing functions G_{N+1}, G_N and G_{HPT}. These functions taken one by one are generally smoother but may still present discontinuities in some cases. If the root of G_{HPT} is smaller than the root of G_{N+1} then there is no solution. Otherwise, the second root to be kept is the smallest one between the root of G_{HPT} and the root of G_N. The algorithms corresponding to the events E_1, E_2 and E_3 are respectively denoted by A_{E_1}, A_{E_2} and A_{E_3}.

3.4. Numerical comparison of the three LS algorithms

The numerical tools used to perform the minimization and the roots search are taken from the Python optimization package in SciPy (`scipy.optimize`). The minimization is conducted with the "`bounded`" algorithm which uses the Brent method to find a local minimum in an interval. The root search is conducted either with the "`toms748`" algorithm [23] in A_{E_1} and A_{E_2} or the "`newton`" algorithm in A_{E_3} as the functions involved are more regular. Each algorithm must be able to treat the three cases mentioned in Section 3.2. The first two cases are frequent while the third case is rarer. Depending on which algorithm is used, the efforts needed to find the roots (or to find out that there is no root) to deal with each case will differ. This is presented in Table 2 where "Minimization" corresponds to the search of a minimum either for $G_{\max}$ or G_{prod}, "Roots search" corresponds to the number of times a root is searched and "G_{HPT} evaluations" is the number of evaluations of G_{HPT} at a given root.

From Table 2, one can expect the algorithm A_{E_3} to be the least demanding in number of code evaluations and the algorithm A_{E_2} to be the most demanding one. This conjecture is verified by estimating 500 failure probabilities corresponding to different input probabilistic distribution parameters with each LS algorithm for a total number of iterations $n_{\text{LS}} = 1 \times 10^3$. The averages $\overline{\#}G_{N+1}$-calls, $\overline{\#}G_N$-calls and $\overline{\#}G_{\text{HPT}}$-calls of the number of evaluations $\#G_i$ of each single limit-state function are calculated as well as the average time $\bar{t}$ required for estimating one failure probability. The number of evaluations that come from the MCF algorithm used for both the LS method and the IS method is not given here as it is the same for all the methods and as it is negligible compared to the total number of evaluations. The results are presented in Table 3 and confirm what was expected from Table

Table 2: Operations performed for each event.

Algorithm		Minimization	Roots search	G_{HPT} evaluations
A_{E_1}	Case 1	1	2	0
	Case 2	1	0	0
	Case 3	1	2	0
A_{E_2}	Case 1	1	2	2
	Case 2	1	1	1
	Case 3	1	3	2
A_{E_3}	Case 1	0	2	2
	Case 2	0	1	1
	Case 3	0	3	2

2. A_{E_3} requires more than three times less total evaluations than A_{E_2} and

Table 3: Performances of each LS algorithm ($A_{E_1}, A_{E_2}, A_{E_3}$).

	A_{E_1}	A_{E_2}	A_{E_3}	IS
$\overline{\#}G_{N+1}$-calls	19162	22524	8484	5×10^4
$\overline{\#}G_N$-calls	19162	22524	3062	5×10^4
$\overline{\#}G_{\mathrm{HPT}}$-calls	19162	1516	996	5×10^4
$\sum \overline{\#}G_i$	57486	46564	13530	1.5×10^5
$\bar{t}(s)$	4.13	3.90	2.21	6.10

more than four times less than A_{E_1}. The ratios in terms of computational time are not the same as for the number of total evaluations as it depends on other factors such as the different functions that are used. Nevertheless, one failure probability estimation with A_{E_3} seems to be almost twice as fast as the two others. It cannot yet be said if the LS algorithms are more efficient than the IS algorithm as it depends on the numbers of iterations n_{LS} and n_{IS}. What can be said is that one IS iteration implies (in the current version of FISTARR) three limit-state evaluations (one for each single limit-state function) and one LS iteration implies roughly an average of 57 total evaluations with A_{E_1}, 46 total evaluations with A_{E_2} and 13 total evaluations with A_{E_3}.

The comparison is now made by looking at the evolution of the estimated failure probability using the three proposed LS algorithms with the IS algorithm and the reference value, denoted as "IS ref." obtained by performing the MCF importance sampling with $n_{\mathrm{IS}} = 10^6$ samples. The IS algorithm

is performed using OpenTURNS, an open-source Python library [19]. The results are presented in Figure 5. The comparison is presented on four dif-

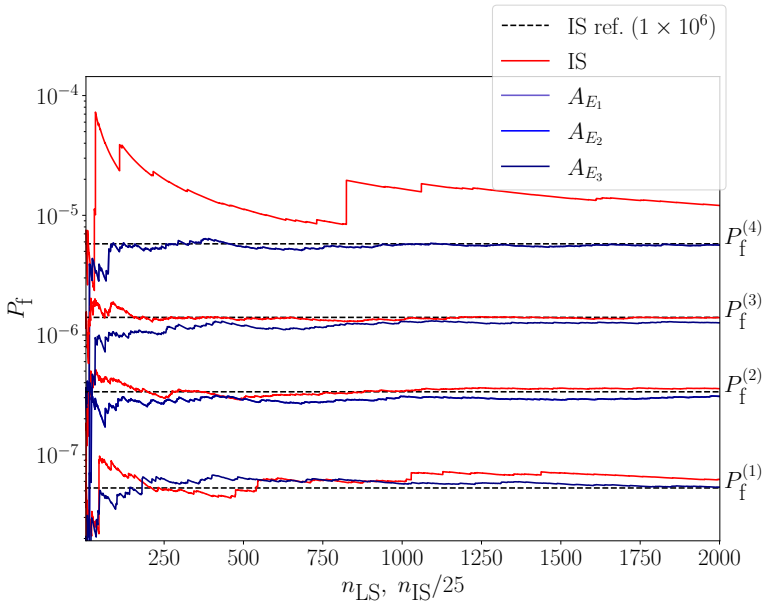


Figure 5: Comparison of P_f estimation obtained with each LS algorithm and the IS algorithm.

ferent configurations of penstocks. The abscissa axis represents the number of LS iterations n_{LS} and the number of IS iterations n_{IS} divided by 25. This means, for example, that 2×10^3 LS iterations correspond to 5×10^4 IS iterations (where one IS iteration evaluates each limit-state function once). One first interpretable result is that the three LS algorithms give identical curves for all four probabilities. The seed of the random generator being the same, this means that the three algorithms find exactly the same roots which is what is expected. The comparison with the IS curve shows that the three LS algorithms are efficient. Indeed, while the IS curve seems to converge rapidly towards the reference value for some configurations (see the middle probabilities), it also seems that the convergence is slower in some cases (see the lowest curve and especially the highest curve). From this initial comparison whose number of samples is too small to draw any final conclusion, the LS algorithms seem relevant for applying info-gap. In the following, A_{E_3} is the only LS algorithm kept as it performs faster. The following section strengthens the comparison by analysing robustness curves obtained using

the different failure probability estimators.

4. Methodology for robustness evaluation

4.1. Info-gap method applied to the reliability of penstocks

Robustness analysis is of particular interest in engineering applications. Classically, a system is considered robust if small variations in an expected state of operation do not considerably deteriorate the expected performance. A robust solution may be preferable over a non-robust optimal solution. The info-gap framework [5] aims at quantitatively measuring the notions of robustness and opportunity in the context of decision making by introducing the following robustness function h^*_{IG} and opportuneness function β^*_{IG}:

$$h^*_{\mathrm{IG}} = \max_h \left\{ \max_{\mathbf{u}\in U\left(h,\widetilde{\mathbf{u}}\right)} R\left(\mathbf{q},\mathbf{u}\right) \leq r^{\mathrm{cr}} \right\} \tag{11}$$

$$\beta^*_{\mathrm{IG}} = \min_h \left\{ \min_{\mathbf{u}\in U\left(h,\widetilde{\mathbf{u}}\right)} R\left(\mathbf{q},\mathbf{u}\right) \leq r^{\mathrm{rw}} \right\} \tag{12}$$

where $h \in \mathbb{R}^+$. Robustness is therefore defined as the maximum amount of uncertainty that can be tolerated, i.e. for which the worst possible performance is still acceptable while opportunity is defined as the minimum amount of uncertainty needed for a reward performance to become possible. Examples of robustness and opportunity curves are shown in Figure 6 (right). The notion of opportunity applied to small failure probabilities is less relevant for safety assessment than the notion of robustness. Therefore, it will not be further discussed in this paper.

Three components appear in the info-gap robustness function in Eq. (11):

- the performance function $R\left(\mathbf{q},\mathbf{u}\right)$ that evaluates the quantity of interest of a system of characteristic vector $\mathbf{q}$ at specific values of the uncertain vector $\mathbf{u}$;
- the critical performance $r^{\mathrm{cr}} \in \mathbb{R}$ which is the value that the quantity of interest must not exceed. Its value may be determined or not in an info-gap analysis;

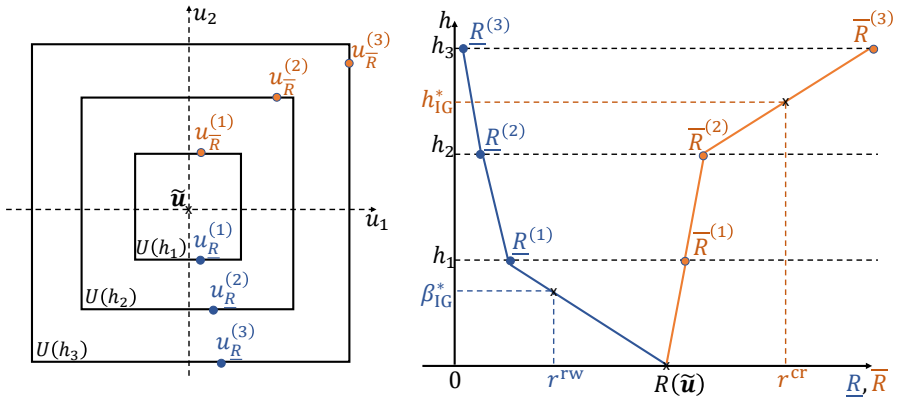


Figure 6: Example of nested convex sets (left) and associated robustness and opportuneness curves (right).

- the uncertainty model $U\left(h, \widetilde{\mathbf{u}}\right)$ which is usually a non-probabilistic convex set [24] of horizon of uncertainty $h \in \mathbb{R}^+$ containing the best estimation $\widetilde{\mathbf{u}}$ (nominal value of $\mathbf{u}$) of the uncertain vector $\mathbf{u}$. For $h = 0$, $U\left(h, \widetilde{\mathbf{u}}\right)$ reduces to $\widetilde{\mathbf{u}}$.

A key feature of the convex uncertainty models is that they are nested as shown in the illustrative example depicted in Figure 6 (left):

$$U\left(h_1, \widetilde{\mathbf{u}}\right) \subseteq U\left(h_2, \widetilde{\mathbf{u}}\right) \text{ for } h_1 \leq h_2. \tag{13}$$

Therefore, the robustness function is monotonous with respect to the horizon of uncertainty and to the performance level.

The general formulation given in the previous section can be applied to the reliability analysis of penstocks. Indeed, the performance function in this case is the failure probability $P_{\mathrm{f}}\left(\mathbf{q}, \mathbf{u}\right)$ where $\mathbf{q}$ represents the characteristic vector of one specific penstock and $\mathbf{u} = \boldsymbol{\theta}$ the vector of uncertain distribution parameters. The critical performance is a target failure probability $P_{\mathrm{f}}^{\mathrm{cr}}$ that must not be exceeded. Its value may be determined before the robustness analysis but may also be chosen after having built the robustness curve.

The uncertain distribution parameters vector $\boldsymbol{\theta}$ considered in this study is $\boldsymbol{\theta} = \left[\mu_{\Delta e_{\mathrm{corr}}}, \mu_{\Delta e_{\mathrm{extra}}}, a_{\max}, \beta_{K_{\mathrm{IC}}}\right]^\top$. It is noted that the choice of epistemic variables is illustrative in order to apply the info-gap framework. The uncertainty model $U\left(h, \widetilde{\boldsymbol{\theta}}\right)$ considered is the basic hyperrectangle convex model defined

as the Cartesian product of all intervals of each uncertain parameter. For a given horizon of uncertainty h, the interval of the parameter θ_i is defined as $I_{\theta_i} = \left[\widetilde{\theta}_i\,(1-h)\,, \widetilde{\theta}_i\,(1+h)\right]$ if $\widetilde{\theta}_i$ is non-negative or $I_{\theta_i} = [1-h, 1+h]$ otherwise. Moreover, the nominal values are set to $\widetilde{\boldsymbol{\theta}} = [1, 0, 4, 90]^\top$. The robustness function in Eq. (11) can be rewritten as follows:

$$h^*_{\mathrm{IG}} = \max_h \left\{ \max_{\boldsymbol{\theta} \in U\left(h, \widetilde{\boldsymbol{\theta}}\right)} P_{\mathrm{f}}\left(\boldsymbol{\theta}\right) \leq P_{\mathrm{f}}^{\mathrm{cr}} \right\}. \tag{14}$$

4.2. Comparison of robustness curves

This section presents the methodology used to estimate the robustness curve of a given nominal configuration of penstock. When applying info-gap in practice, it is generally not necessary to solve the double optimization problem as in Eq. (14) in order to find the unique value h^*_{IG}. Instead, it is less time consuming and more informative to estimate the robustness curve by estimating the highest failure probability $\overline{P_{\mathrm{f}}}(h_i) = \max_{\boldsymbol{\theta} \in U\left(h_i, \widetilde{\boldsymbol{\theta}}\right)} P_{\mathrm{f}}\left(\boldsymbol{\theta}\right)$ at the discretized values $h_i \in [0; h_{\max}]$. Therefore, it is up to the analyst to choose the number of horizons of uncertainty and its maximum value.

For this application, it has been chosen to construct the robustness curves with 10 values $h_i \in [0; 0.2]$. To find the maximum of the failure probability, the global optimization algorithm "`GN_ORIG_DIRECT_L`" adapted from the original `DIRECT` algorithm [25, 26] and implemented in the `NLopt` Python library is used with a maximum number of 500 evaluations. This algorithm searches for the optimal value by iteratively dividing the hypercube optimization space into optimal hyperrectangles.

The objective is to compare on one nominal configuration of penstock the robustness curves obtained using the A_{E_3} and IS algorithms as failure probability estimators with the reference curve. The comparison is made in terms of robustness curves (see Figure 7), relative error $\eta_{P_{\mathrm{f}}}$ with the reference curve (see Figure 8) and cumulative calculation time (see Figure 9) considering $n_{\mathrm{IS}} \in \{2.5 \times 10^4, 5.0 \times 10^4, 1.0 \times 10^5\}$ and $n_{\mathrm{LS}} \in \{1.0 \times 10^3, 2.0 \times 10^3, 3.0 \times 10^3\}$.

It appears from Figures 7 and 8 that the IS algorithms perform rather well in general, except for the three last values of h for which the robustness curves deviate from the reference curve. However, even in this less favorable case, the optimization process will automatically provide conservative

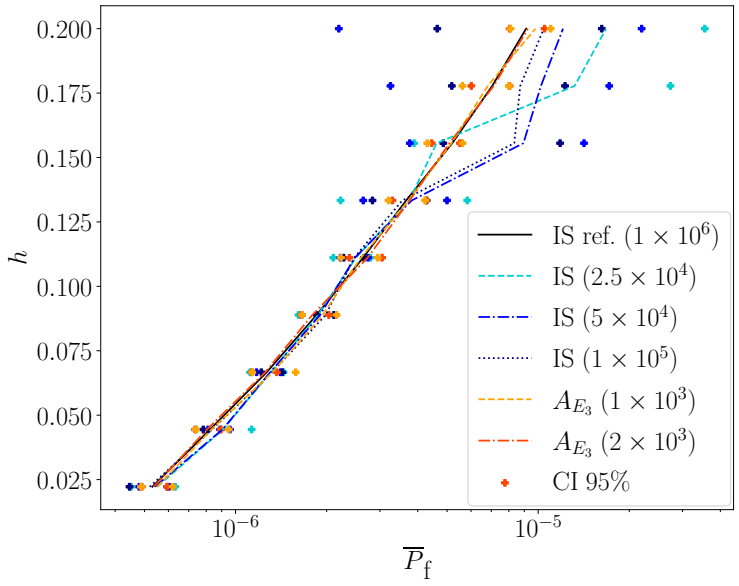


Figure 7: Comparison of robustness curves obtained with the proposed LS algorithm A_{E_3} and the IS algorithm.

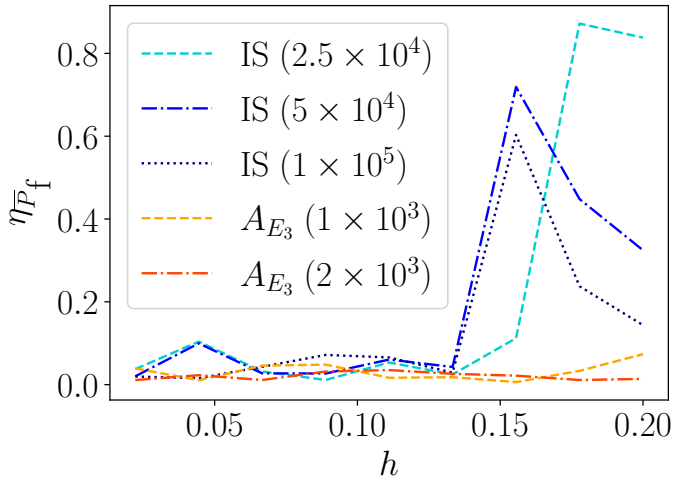


Figure 8: Comparison of the error $\eta_{\overline{P}_{\mathrm{f}}}$ of each algorithm w.r.t. the reference robustness curve.

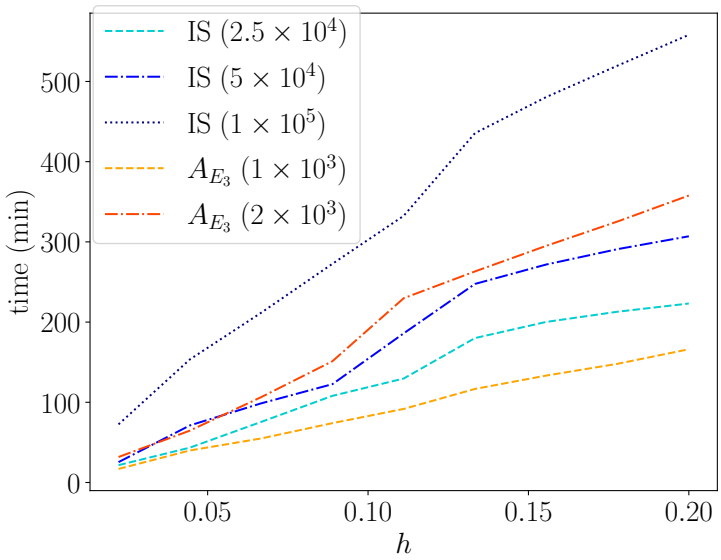


Figure 9: Cumulative computational time for obtaining the robustness curves.

results ($\overline{P}_{\mathrm{f}}$ is overestimated) and the confidence intervals contain the reference curve: their performance remains acceptable even in this case. Note that these non adaptive IS algorithms could be further optimized; moreover, other optimization algorithms than LD_MMA in the multi-constraint design point may converge better. This deviation does not seem to happen with the proposed algorithm A_{E_3} for which the robustness curves remain close to the reference curve. Moreover, the proposed LS algorithms seem also efficient in terms of cumulative computational time as shown in Figure 9. Although no definitive conclusion can be made regarding the comparative efficiency of the proposed algorithms with the IS ones, the adapted LS algorithms seem to be well suited in this context of robustness analysis.

4.3. Synthesis

Drawing robustness curves requires an efficient failure probability estimator over the whole uncertainty space. Indeed, as optimization is performed repeatedly, only a few bad estimations suffice to make the robustness curve deviate. In addition to a general trend to provide conservative results that could be observed from FORM-IS in standard penstock reliability evaluations, the fact that the optimization algorithm used is global and that it

searches for a maximum value will generally tend to make the errors conservative which is preferred for safety assessments. Moreover, a criterion based on the coefficient of variation of each failure probability estimation could be used to insure a sufficient convergence at each evaluation.

However, the info-gap robustness analysis is therefore very instructive on the efficiency of the failure probability estimator in the considered uncertainty space. The proposed adapted line sampling algorithms represent an interesting alternative as they manage to better target the restricted failure domain and correctly estimate the roots. In this reliability application, the partial knowledge on the behaviors of each limit-state function is helpful for adapting the root search. Nevertheless, each LS iteration still requires a large number of G-functions evaluations. The following section presents a method that aims at training artificial neural networks (ANN) in order to predict the roots for any sampled line and for any uncertain vector $\boldsymbol{\theta}$ during the robustness analysis.

5. Combination of two artificial neural networks for the LS roots prediction

5.1. Problem statement

As the G-functions involved in Eq. (3) are a series and a combination of analytical expressions, a single failure probability estimation may be obtained within a few seconds. Nevertheless, when considering no information that could simplify the optimization process such as a monotonous behavior with respect to the epistemic distribution parameters, applying the info-gap method as it is done in the previous section requires the evaluation of several thousands of failure probabilities. Therefore, being able to reduce the computational time of one probability evaluation remains relevant. To do so, the literature offers a wide variety of methods in the case of parametric p-boxes [27]. Among them, some methods aim at substituting the expensive G-function with a surrogate model [28, 29]. This is not relevant in the present application as the G-functions are not expensive to evaluate. It is considered here to use the surrogate models in order to directly evaluate the existence and (when they exist) the values of the roots for any joint vector $(\mathbf{u}, \boldsymbol{\theta})$. As the G-functions are relatively fast to compute, several thousand training samples may be considered for building predictive surrogate models. Therefore, the choice made in the present paper is to use ANNs rather

than other types of surrogate models such as kriging or polynomial chaos expansions.

Other methods exist such as those based on evaluating many failure probabilities using a unique input dataset. In [30, 31], failure probabilities (or more precisely predicted failure probabilities in [31]) are estimated from a unique set of samples generated in the augmented space $(\mathbf{X}, \boldsymbol{\Theta})$ where $\boldsymbol{\Theta}$ is an instrumental probabilistic distribution on the distributional parameters $\boldsymbol{\theta}$. However, the mathematical formulations involved in this method are expected to increase the computational time in the present application. In [32], the line sampling roots obtained given one input distribution parameters are transformed in order to estimate failure probabilities with different input distribution parameters without having to search for the new roots. Nevertheless, this method does not seem directly applicable in the present case where some iterations do not have any roots. In [33], the method called "weighted importance sampling" (WIS) enables to evaluate failure probabilities from a single dataset very easily as it only requires the calculation of a ratio of densities. This method is also considered in this work and will be part of the comparison when computing robustness curves. Its application is shortly described in Appendix A.

5.2. Generalities on artifical neural networks

This part does not aim at giving an extended description of ANNs but only at presenting the basic notions necessary to understand how they may be of use for reliability analysis. ANNs represent a mathematical structure that processes information from an input layer to an output layer through hidden layers [34]. The information is passed from one layer to another with some specific functions called "artificial neurons" as illustrated in Figure 10. Each neuron belonging to the layer $l^{(i)}$ receives as an input, a linear combination on the outputs $s_k^{(i-1)}$ of the neurons of the previous layer $l^{(i-1)}$ with weights $w_{k,j}^{(i)}$ and a bias term $b_j^{(i)}$. The input is then processed with an activation function $f^{(i)}$ whose output $s_j^{(i)}$ is passed to the neurons of the next layer. This simple mechanism is depicted in Figure 10. In this paper, fully connected feedforward ANNs are considered which simply corresponds to architectures where the information only goes from all the neurons of layer $l^{(i-1)}$ to all the neurons of layer $l^{(i)}$ (but not between neurons of a same layer which is the case for recurrent neural networks).

An ANN may learn complex relationships between inputs and outputs by

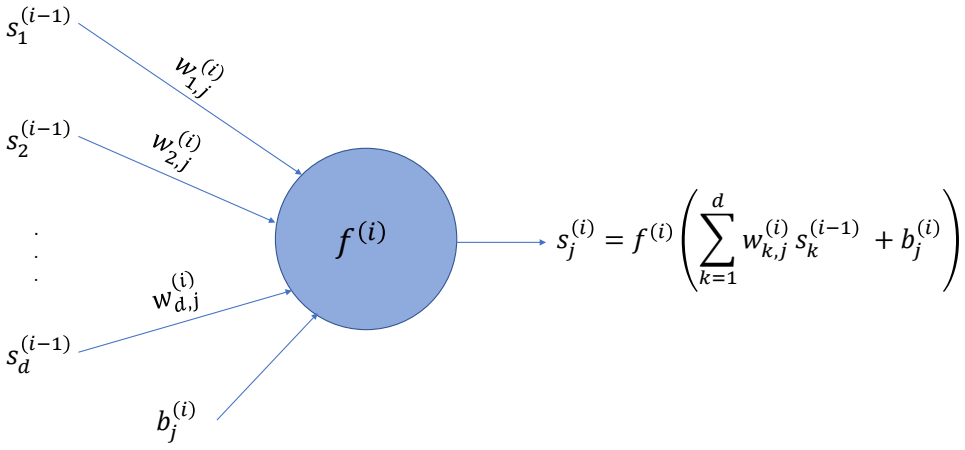


Figure 10: Representation of a single artificial neuron.

training it with an available dataset (e.g., composed of inputs-output realizations). Indeed, by performing backpropagation through gradient descent [35], the ANN is able to update the values of the weights $w_{k,j}$ and the biases b_j such that the errors (defined through a loss function) between the output dataset and the ANN outputs are minimized. The application range of such networks is very wide as ANNs may be used for classification and regression problems. Moreover, it is able to treat all sorts of information [36]. Therefore, ANNs also find their use in reliability analyses as surrogate models, most often to replace an expensive limit-state function. In [37], [38] and [39], ANNs are combined with Monte Carlo simulation, subset simulation and line sampling respectively. A review of their use in the context of reliability analysis is proposed in [40]. In the present work, ANNs are combined with LS in order to directly predict the roots associated to each $\mathbf{u}_{\boldsymbol{\alpha}}^{\perp}$ drawn from the LS algorithm.

5.3. Proposed methodology based on artificial neural networks

ANNs are combined to the LS-based A_{E_3} algorithm, which has been identified as the most efficient in Subsection 4.2, in order to reduce the computational time required for obtaining a robustness curve. What makes the A_{E_3} still time consuming is that it requires a large number of evaluations of the

G-functions, first to assess the existence of roots and, second, to evaluate their values. The objective here is to be able, for any joint sample $(\mathbf{u}, \boldsymbol{\theta})$, to predict the answers of the two previous problems based on training samples $(\mathbf{u}, \boldsymbol{\theta})^{\text{train}}$. The fact of considering the probabilistic standard vector $\mathbf{u}$ together with the epistemic uncertain vector $\boldsymbol{\theta}$ as the input of the ANN enables to create a single surrogate model applicable during the whole info-gap analysis.

Two types of ANNs are jointly proposed. First, a classification ANN, denoted by ANN_1, is necessary in order to predict if roots exist or not for a given sample $(\mathbf{u}, \boldsymbol{\theta})^{(i)}$. Then, a regression ANN, denoted as ANN_2, is used to predict the values of both roots when they exist. The procedure that is followed to build both ANNs is presented hereafter:

1. Generate n_{train} training and n_{val} validation samples of $\mathbf{u}$ according to the independent standard Gaussian distribution ;
2. Generate n_{train} training and n_{val} validation samples of $\boldsymbol{\theta}$ according to the uniform distribution with the bounds $\left[\underline{\boldsymbol{\theta}}\left(h_{\max}\right), \overline{\boldsymbol{\theta}}\left(h_{\max}\right)\right]$ with a user-defined $h_{\max}$;
3. For each $\left(\mathbf{u}^{(i)}, \boldsymbol{\theta}^{(i)}\right)^{\text{train}}$ and $\left(\mathbf{u}^{(i)}, \boldsymbol{\theta}^{(i)}\right)^{\text{val}}$, assess the existence or not of roots using algorithm A_{E_3}. Any joint vector for which no root exists is denoted $(\mathbf{u}, \boldsymbol{\theta})^{(0)}$ while the others are denoted $(\mathbf{u}, \boldsymbol{\theta})^{(1)}$;
4. Build and learn the surrogate model ANN_1 with the samples $(\mathbf{u}, \boldsymbol{\theta})^{\text{train}}$ and validate its performance with the samples $(\mathbf{u}, \boldsymbol{\theta})^{\text{val}}$;
5. Build and learn the surrogate model ANN_2 with the samples $(\mathbf{u}, \boldsymbol{\theta})^{(1),\text{train}}$ and validate its performance with the samples $(\mathbf{u}, \boldsymbol{\theta})^{(1),\text{val}}$.

A lot of parameters may be tuned when building ANNs such as the number of layers, the number of neurons per layer or the type of loss and accuracy metrics. Both ANNs are built using the Python libraries Keras and Tensorflow. More information about the architectures and parameters of ANN_1 and ANN_2 for the penstock use-case is given in Appendix B.

A small number of errors in the roots classification may lead to unfixable errors especially as less compensation will take place with highly efficient ANNs. One way of treating this issue is to consider a multi-fidelity approach and to combine the classification surrogate model ANN_1 with the initial algorithm A_{E_3} for estimating $P_{\text{f}}\left(\boldsymbol{\theta}\right)$. Indeed, for a given joint vector $(\mathbf{u}, \boldsymbol{\theta})$, the output of ANN_1 corresponds to the probability that roots exist. Therefore, if the output is close to 0, one can have strong confidence that no root exists.

On the contrary, if the output is close to 1, one can have strong confidence that some roots exist. However, if the output takes a value close (where close might be quantified by the analyst) to 0.5, then one might want to check the correct answer with the A_{E_3} algorithm. Consequently, by defining the security value $s \in [0, 0.5]$, the hybrid multi-fidelity method denoted by "A_{E_3}-ANN" is proposed. Thus, it simply adds the following operation:

- if the output $\text{ANN}_1\left(\mathbf{u}^{(i)}, \boldsymbol{\theta}^{(i)}\right) \in [0.5 - s; 0.5 + s]$, one can estimate the roots using A_{E_3} ;
- otherwise, one can reasonably trust the result obtained from ANN_1.

The complete procedure is depicted in Figure 11.

6. Application cases

6.1. Rosenbrock function

The methodology is first applied to a limit-state function based on the Rosenbrock function in two dimensions:

$$g\left(X_1, X_2\right) = 100\left(X_2 - X_1^2\right)^2 + \left(X_1 - 1\right)^2 - 0.01 \tag{15}$$

where $X_1 \sim N\left(\theta_1, 1\right)$ and $X_2 \sim N\left(\theta_2, 1\right)$. Indeed, this numerical case has a similar problematic as the penstock reliability problem in terms of root search. As depicted in Figure 12.(a) for $\theta_1 = 1.5$ and $\theta_2 = 0$, the limit-state function takes on a very narrow elliptic shape. Due to this geometry and to the fact that the limit-state function is not formulated as an intersection, the LS algorithm A_{E_1} is best suited. However, Figure 12.(b) shows that a large number of LS iterations and IS samples are needed in order to converge to the reference failure probability.

The ANN-based methodology is applied by considering $n_{\text{train}} \in [1 \times 10^3, 3 \times 10^4]$, $n_{\text{val}} = 0.2 n_{\text{train}}$, $\Theta_1 \sim U\left(0, 3\right)$ and $\Theta_2 \sim U\left(-1, 1\right)$. The four following metrics relevant to the performance of ANN_1 and ANN_2 are defined:

- "false root" is the proportion of wrongly declared existing roots from ANN_1 ;
- "forgotten root" is the proportion of existing roots forgotten by ANN_1 ;

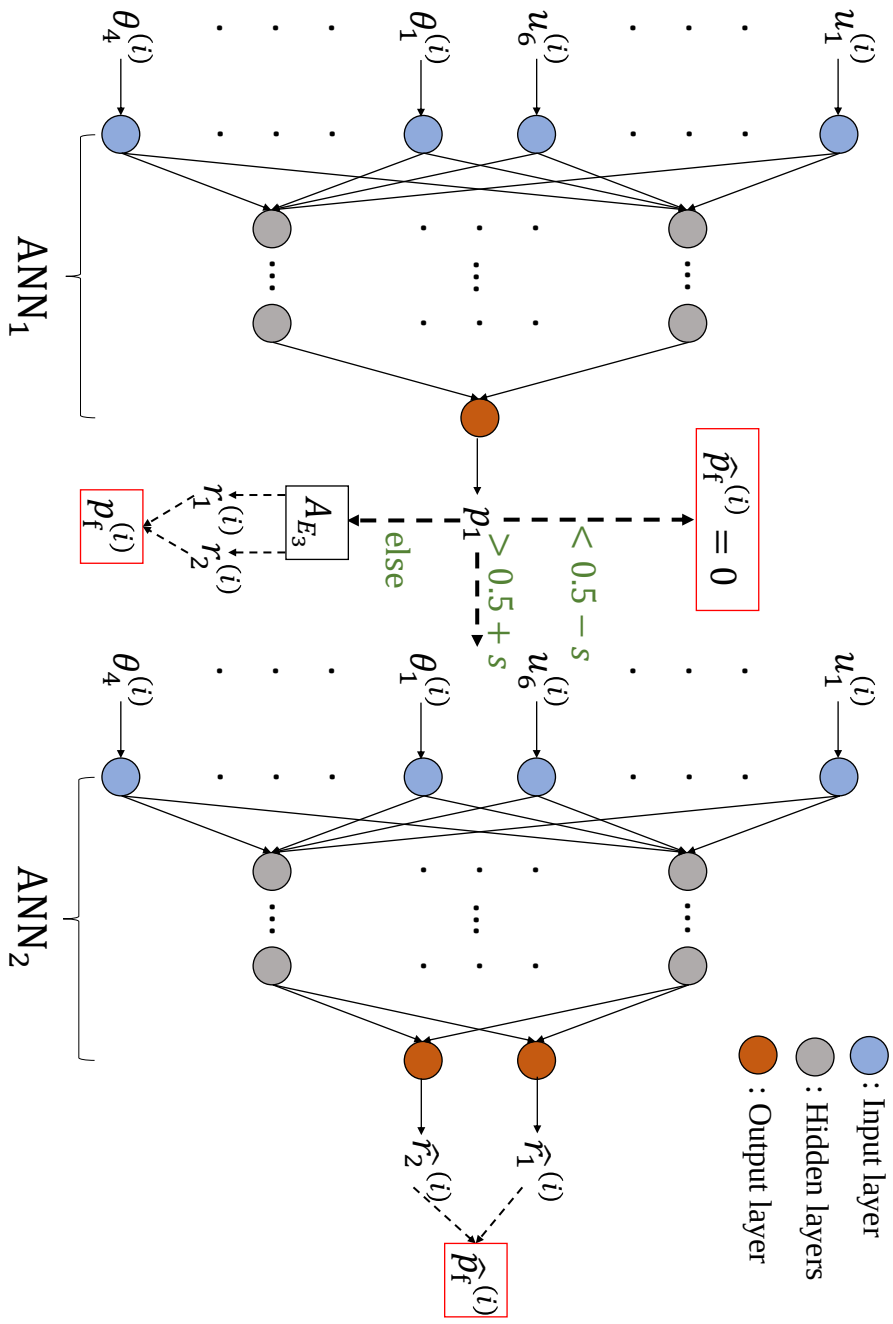
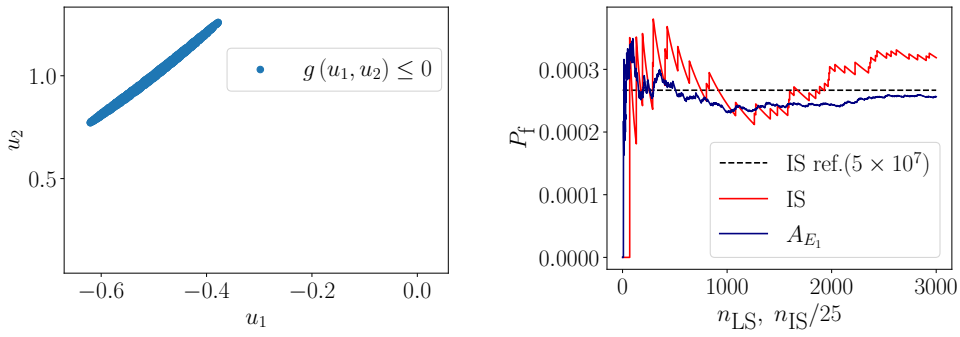


Figure 11: Illustration of the methodology combining ANNs and A_{E_3} for the reliability of penstocks.



(a) (b)

Figure 12: Illustration of a limit-state function (a) and comparison of the evolution of P_f with A_{E_1} and with importance sampling (b) for the Rosenbrock function.

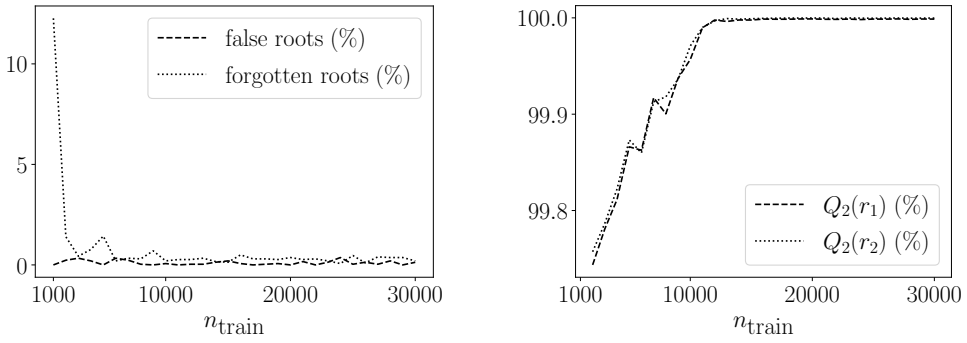
- $Q^2(r_1)$ is the coefficient of preditivity calculated on the first root r_1 in common between ANN_1 and A_{E_1} ;
- $Q^2(r_2)$ is the coefficient of preditivity on the second root r_2 in common between ANN_1 and A_{E_1}

where the coefficient of predictivity has the following expression:

$$Q^2(r_i) = 1 - \frac{\sum_{j=1}^{n_{r_i}} \left(r_i^{(j)} - \widehat{r}_i^{(j)} \right)^2}{\sum_{j=1}^{n_{r_i}} \left(r_i^{(j)} - \overline{r_i}^{(j)} \right)^2} \tag{16}$$

where n_{r_i} is the number of real roots predicted by ANN_1, $r_i^{(j)}$ the roots obtained with A_{E_1}, $\widehat{r}_i^{(j)}$ the roots obtained with ANN_2 and $\overline{r_i}^{(j)}$ the mean on all roots.

The values of these metrics are calculated on 3×10^3 new testing samples $(\mathbf{u}, \boldsymbol{\theta})^{\text{test}}$. Note that the availability of such testing samples is not always present for more time-demanding applications. Figure 13 presents the impact of the number of training samples on the four metrics. Except for $n_{\text{train}} = 1000$ where the proportion of forgotten roots is high (actually is is equal to the true proportion of existing roots meaning that ANN_1 misjudged every single existing root), the proportions of wrong classifications quickly become very low (typically lower than 1%). It also appears that the coefficients of



(a) (b)

Figure 13: Illustration of the performances of ANN_1 (a) and ANN_2 (b) on testing samples in function of the number of training samples on the Rosenbrock function.

predictivity of both roots are very high (typically greater than 99%) even for $n_{\text{train}} = 1000$.

The methodology is now tested with $n_{\text{train}} = 3 \times 10^4$ at three randomly chosen uncertain vectors $\boldsymbol{\theta}^{(i)}$: $\boldsymbol{\theta}^{(1)} = [2.37, 0]^\top$, $\boldsymbol{\theta}^{(2)} = [0.84, -0.99]^\top$ and $\boldsymbol{\theta}^{(3)} = [0.88, -0.08]^\top$. One can see in Figure 14.(a) that ANN_1 manages very well to predict the orthogonal points for whtch roots exist and that ANN_2 is very precise on the estimation of $r_{1,2}$ although the zoom in Figure 14.(b) seems to show that the elliptic shape is simplified by two lines. These good visual performances are confirmed with the comparison of the failure probabilities estimations presented in Figure 15. In this case, the proportions of forgotten roots (0.44%, 0.1% and 0.18%) and the proportions of false roots (0.02%, 0.04% and 0%) are very low such that there is no need to apply the security value s.

6.2. Reliability of penstocks

The ANN methodology is now applied to the reliability assessment of penstocks. As already defined in Subsection 4.1, the vector of uncertain distribution parameters is $\boldsymbol{\theta} = [\mu_{\Delta e_{\text{corr}}}, \mu_{\Delta e_{\text{extra}}}, a_{\max}, \beta_{K_{\text{IC}}}]^\top$ with the following nominal vector $\widetilde{\boldsymbol{\theta}} = [1, 0, 4, 90]^\top$. The maximum value of the horizon of uncertainty having been set at $h_{\max} = 0.2$, the training is performed considering $\Theta_1 \sim U(0.8, 1.2)$, $\Theta_2 \sim U(-0.2, 0.2)$, $\Theta_3 \sim U(3.1, 4.9)$ and $\Theta_4 \sim U(71, 109)$. Again, the values of the four testing metrics are calculated on 3×10^3 new testing samples for $n_{\text{train}} \in [1 \times 10^3, 3 \times 10^4]$ and

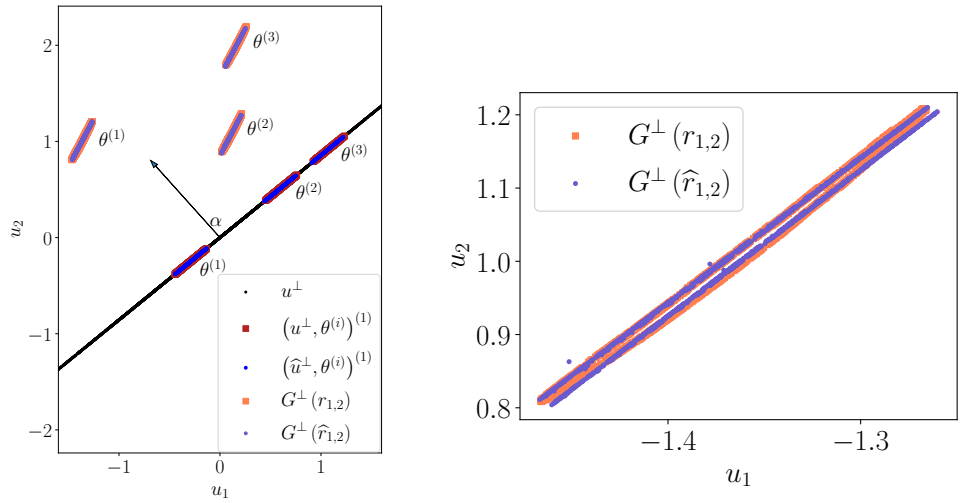


Figure 14: Illustration of the performances of ANN_1 and ANN_2 for three distribution parameters vectors (a) with a zoom on one limit-state (b) for the Rosenbrock function.

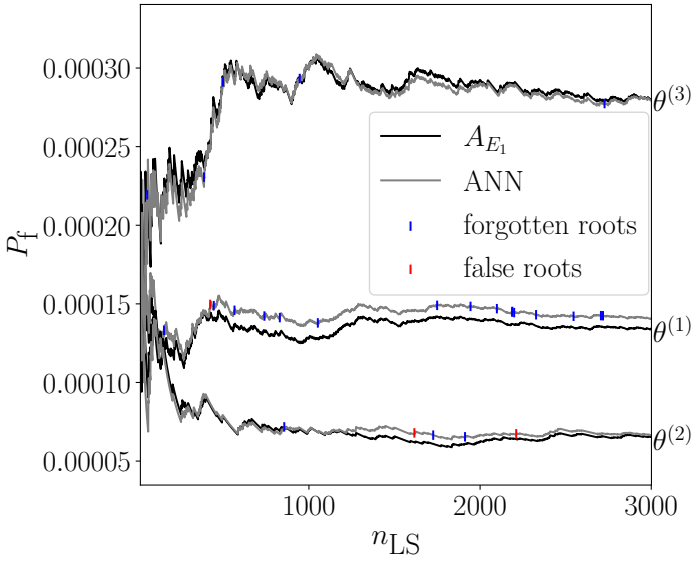
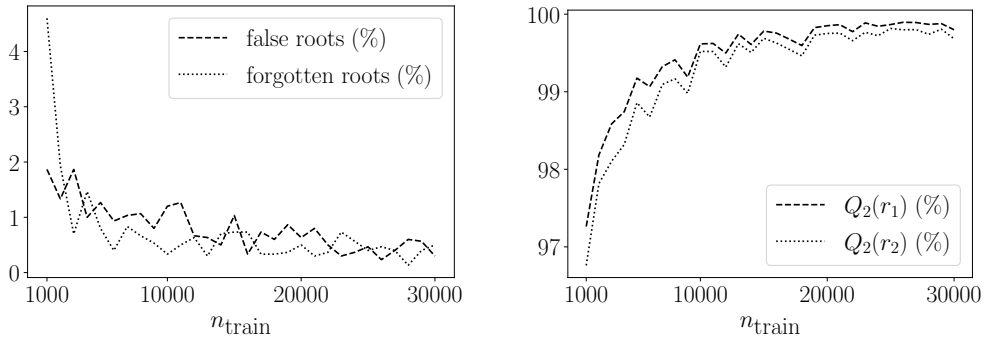


Figure 15: Comparison of the evolution of P_{f} with A_{E_1} and the ANNs for three distribution parameters vectors for the Rosenbrock function.

$n_{\text{val}} = 0.2n_{\text{train}}$. The results shown in Figure 16 reveal good performances but for a higher number of training samples compared to the results obtained on the Rosenbrock function. One may notice that the coefficient of



(a) (b)

Figure 16: Illustration of the performances of ANN_1 (a) and ANN_2 (b) on testing samples in function of the number of training samples on the penstock use-case.

predictivity related to r_2 is always lower than the one related to r_1. A possible explanation is that the first root always corresponds to the limit-state function G_{N+1} whereas the second root either corresponds to G_N (in most cases) or to G_{HPT} which may be a more challenging feature to understand for ANN_2.

The procedure is now tested on the two samples $\boldsymbol{\theta}^{(1)} = [1,\ 0,\ 4,\ 90]^\top$ and $\boldsymbol{\theta}^{(2)} = [1.2,\ -0.2,\ 4.8,\ 108]^\top$ as they both represent the nominal point and one of the vertex points respectively. The results with the A_{E_3} algorithm and the ANNs are compared for $n_{\text{train}} = 3 \times 10^4$. The values of the four testing metrics are given in Table 4 for $n_{\text{LS}} = 3 \times 10^3$.

Table 4: ANNs metric values on $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$.

	false roots (%)	forgotten roots (%)	$Q^2(r_1)$ (%)	$Q^2(r_2)$ (%)
$\boldsymbol{\theta_1}$	0.16	0.63	99.8	99.7
$\boldsymbol{\theta_2}$	1.6	0.06	99.7	99.7

Figure 17 compares the values of $p_{\text{f}}^{(i)}$ obtained from the ANN and from A_{E_3} at both distribution parameters vectors $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$. The comparisons

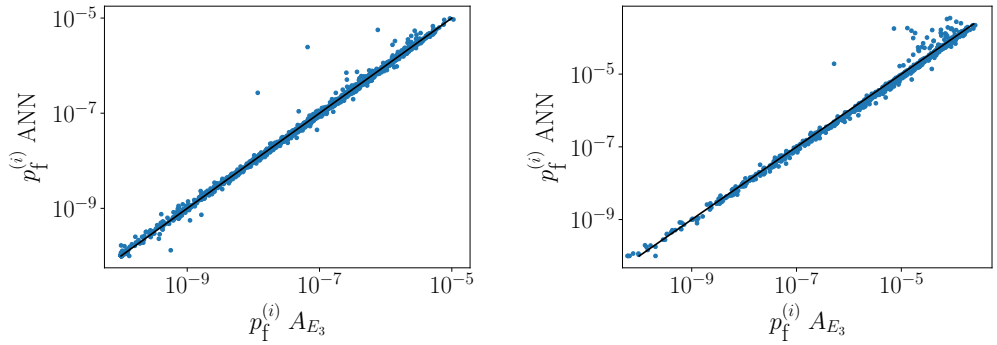


Figure 17: Comparison between the values of $p_{\mathrm{f}}^{(i)}$ from A_{E_3} and from the ANN at $\boldsymbol{\theta}^{(1)}$ (a) and $\boldsymbol{\theta}^{(2)}$ (b).

of the evolution of $P_{\mathrm{f}}\left(\boldsymbol{\theta}^{(1)}\right)$ and $P_{\mathrm{f}}\left(\boldsymbol{\theta}^{(2)}\right)$ with the two algorithms are presented in Figure 18. The evolution of $P_{\mathrm{f}}\left(\boldsymbol{\theta}_1\right)$ with both algorithms is almost

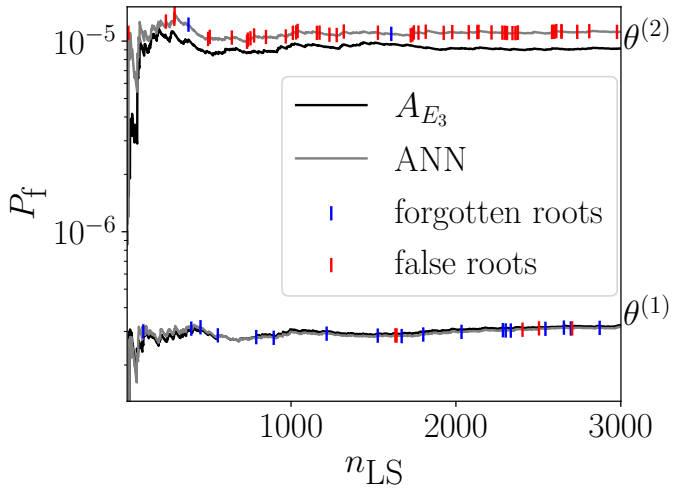


Figure 18: Evolution of $P_{\mathrm{f}}\left(\boldsymbol{\theta}^{(1)}\right)$ and $P_{\mathrm{f}}\left(\boldsymbol{\theta}^{(2)}\right)$ with A_{E_3} and the ANNs.

identical. Indeed the number of wrong classifications from ANN_1 is very low and ANN_2 seem to predict both roots with high precision. The evolution of

$P_{\mathrm{f}}(\boldsymbol{\theta}_2)$ with the ANNs is slightly overestimated which may be caused either by the evaluation of many false roots or by an overestimation of each single $p_{\mathrm{f}}^{(i)}$. It is hard to tell from Figure 17.(b) if the single failure probabilities are overestimated as most predictions seem slightly underestimated but the worst predictions correspond to a few overestimated predictions. However, Figure 18 clearly shows a relatively large proportion (1.6%) of false roots which will automatically increase the estimated failure probability. To improve the estimation of $P_{\mathrm{f}}(\boldsymbol{\theta}_2)$, the security value s may be used to reduce the number of false roots. Figure 19 presents the evolution of $P_{\mathrm{f}}(\boldsymbol{\theta_2})$ for three different values of s. As expected, increasing the value of s brings the

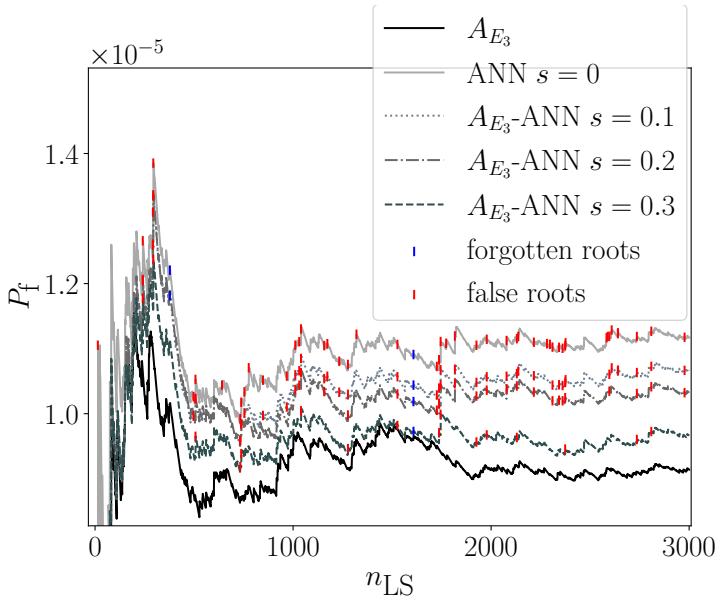


Figure 19: Evolution of $P_{\mathrm{f}}(\boldsymbol{\theta_2})$ for different security values.

probability estimation curve closer to the one obtained with A_{E_3}. The side effect is that increasing s automatically increases the number of G-functions evaluations. However, Figure 20 reveals that a very large proportion of the output of ANN_1 is either very close to 0 or very close to 1 meaning that the verification process remains occasional.

6.3. Application to the estimation of robustness curves

The proposed methodology is used for estimating the robustness curves. The comparison is made by considering the following failure probability es-

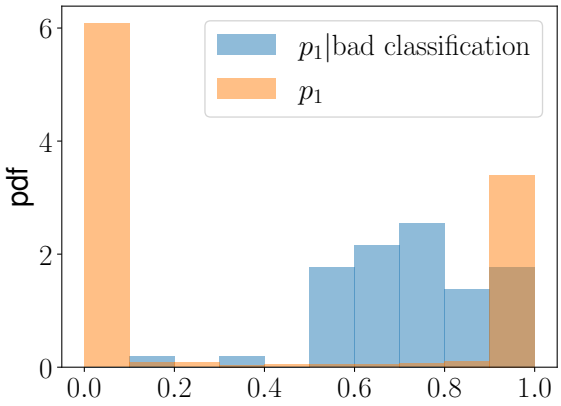


Figure 20: The pdf of the probability output p_1 of ANN_1 compared to its pdf conditioned to bad classifications for $\boldsymbol{\theta}^{(2)}$.

timators: "A_{E_3} 2×10^3" which estimates P_f with A_{E_3} for $n_\text{LS} = 2 \times 10^3$, "$A_{E_3}$-ANN" which uses ANN_1 and ANN_2 applying the security value $s = 0.3$ for $n_\text{LS} \in [1 \times 10^3, 2 \times 10^3]$, "WIS 1×10^5" which uses the Weighted Importance Sampling algorithm in [33] with a unique importance sampling of 10^5 samples considering $\boldsymbol{\theta} = \widetilde{\boldsymbol{\theta}}$ and "IS ref" as the same reference curve as in the previous robustness comparisons. Again, the comparison is made in terms of robustness curves (see Figure 21), error with the reference curve (see Figure 22) and cumulative computational time (see Figure 23).

Figures 21 and 22 reveal accurate estimations of the robustness curves using the ANNs and using the WIS approach. Actually, it shows that using the WIS approach by using a unique importance sampling of 1×10^5 samples reduces the errors that were obtained when repeatedly using the IS algorithm. A further reduction could be achieved by combining WIS with optimized FORM-IS algorithms leading to a lower sample size. In Figure 23, it can be seen how the use of ANNs considerably reduces the computational time even when considering a security value $s = 0.3$.

7. Conclusion

In this paper, two original and complementary methodologies are proposed in order to efficiently apply the info-gap framework to the reliability assessment of penstocks.

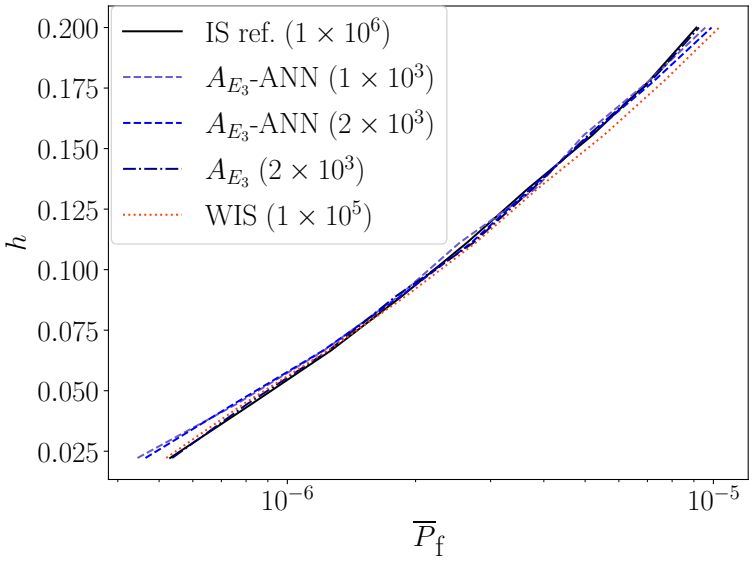


Figure 21: Robustness curves obtained with the A_{E_3}, A_{E_3}-ANN and WIS algorithms.

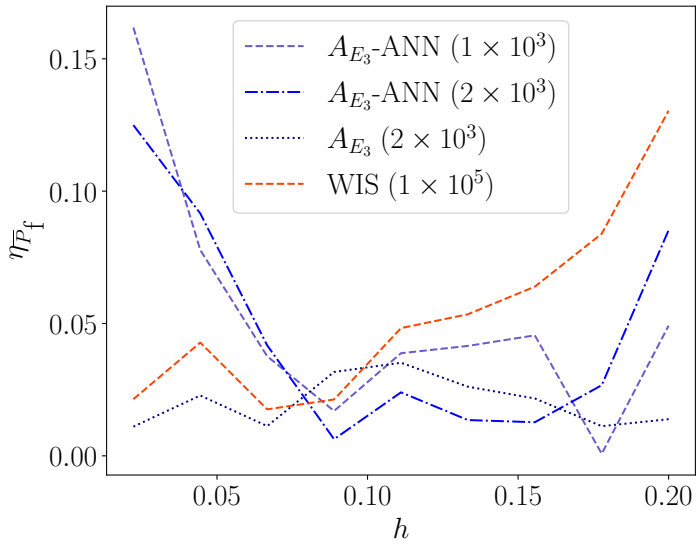


Figure 22: Relative error on the robustness curves obtained with the A_{E_3}, A_{E_3}-ANN and WIS algorithms.

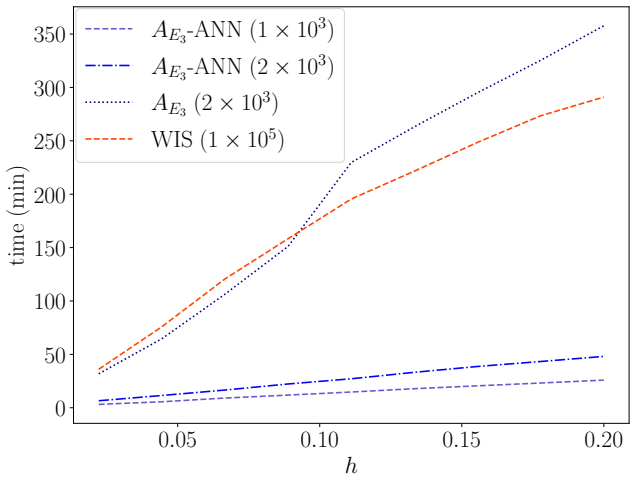


Figure 23: Cumulative computational time for obtaining the robustness curves obtained with the A_{E_3}, A_{E_3}-ANN and WIS algorithms.

Firstly, the inner challenging reliability problem is tackled by the use of three customized LS algorithms (A_{E_1}, A_{E_2} and A_{E_3}) based on three equivalent formulations of a complex limit-state function made of intersection of events. While considering the root search differently, each algorithm yields similar estimations of the failure probability. The good performances of the proposed algorithms are enhanced with their use for estimating robustness curves. However, no definitive conclusion can be drawn about their relative efficiency compared to FORM-IS algorithms that perform rather well and could be further optimized.

Secondly, the root search procedure is improved in terms of computational time with the use of two artificial neural networks. The first one enables to predict the existence (or not) of roots for any given line search and for any value of the epistemic uncertain vector $\boldsymbol{\theta}$. The second one predicts the values of both roots when they exist. Although the ANNs performance metrics are very good, a few bad predictions may lead to non-negligible errors on the failure probability. Therefore, a security value is proposed in order to decide whether the classification ANN should be trusted or whether the initial algorithm should be used. The methodology is then tested and proven to be very efficient for estimating info-gap robustness curves. However, the security value calibrated for the tested configuration may not be generic and a small

remaining estimation bias may exist anyway. Moreover, info-gap robustness analyses consist in local reliability analyses around a nominal configuration, and surrogate-based techniques like ANN may be less appropriate for standard penstock reliability assessments where large variations of parameters are considered.

The use of neural networks is motivated by the fact that the limit-state functions involved in the industrial use case are relatively fast to evaluate which enables a large dataset for the training process. In many practical applications, such a large dataset may not be available due to time-consuming numerical models. In this case, it would be necessary to consider other types of surrogates models such as Gaussian process regression. In particular, methods based on active learning such as in [41] may present a high interest especially if there is a way to apply it in the augmented space which includes the uncertain distribution parameters.

Moreover, the high computational cost for estimating info-gap robustness curves is due to the choice of not making any assumption when successively searching for the maximum failure probability at each horizon of uncertainty h. Valuable information, such as monotonic behavior of the failure probability with respect to distribution parameters, may be assessed from a preliminary study. For example, dedicated sensitivity measures may guide the optimization process especially as they might be obtained simultaneously with the failure probability estimation such as classical FORM importance factors when using FORM-IS or such as in [42] in the context of line sampling.

Appendix A. Presentation of the Weighted Importance Sampling technique

In [33], the objective is to use the failure probability result obtained at a nominal value of the distribution parameters $\widetilde{\boldsymbol{\theta}}$ to estimate failure probabilities at different values $\boldsymbol{\theta}$ without the need to generate new samples. This general framework is referred as weighted approach and is applied to Monte Carlo simulation, importance sampling and subset simulation. In the present paper, only its combination with importance sampling is analyzed.

The basic idea is similar to classical importance sampling. For any $\boldsymbol{\theta}$, the failure probability is expressed as follows

$$P_{\mathrm{f}}(\boldsymbol{\theta}) = \int I_{\mathcal{F}}(\mathbf{x}) \frac{f(\mathbf{x}|\boldsymbol{\theta})}{H(\mathbf{x})} H(\mathbf{x}) d\mathbf{x} \tag{A.1}$$

where $f(\mathbf{x}|\boldsymbol{\theta})$ is the conditional pdf of the random vector $\mathbf{X}$ and $H(\mathbf{x})$ is the importance sampling instrumental pdf to be defined by the user. Eq.(A.1) may be rewritten as an expectation under $H(\mathbf{x})$

$$P_{\mathrm{f}}(\boldsymbol{\theta}) = \mathbb{E}_H \left[I_{\mathcal{F}}(\mathbf{x}) \frac{f(\mathbf{x}|\boldsymbol{\theta})}{H(\mathbf{x})} \right] \tag{A.2}$$

which may be estimated by generating N samples $\mathbf{x}^{(j)}$ from $H(\mathbf{x})$ as follows

$$P_{\mathrm{f}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{j=1}^{N} I_{\mathcal{F}}(\mathbf{x}^{(j)}) \frac{f\left(\mathbf{x}^{(j)}|\boldsymbol{\theta}\right)}{H\left(\mathbf{x}^{(j)}\right)}. \tag{A.3}$$

Therefore, it can be seen that, no matter the value of $\boldsymbol{\theta}$, the indicator function (which is generally expensive to evaluate as it involves the computer model) is calculated with the same samples $\mathbf{x}^{(j)}$ generated from $H(\mathbf{x})$. Only the conditional pdf $f\left(\mathbf{x}^{(j)}|\boldsymbol{\theta}\right)$ needs to be reevaluated for each $\boldsymbol{\theta}$.

The performance of the method highly depends on the choice of the instrumental pdf. Indeed, the estimation in Eq.(A.3) is considered as a local approximation as a better convergence will be achieved for values of $\boldsymbol{\theta}$ that are representative of the instrumental pdf $H(\mathbf{x})$. However, such framework is compatible with the info-gap robustness analysis for relatively small horizons of uncertainty h as maximum failure probabilities are searched for around a nominal value $\widetilde{\boldsymbol{\theta}}$. Therefore, the method is considered for the robustness analysis of the reliability of penstocks. The instrumental pdf is constructed

based on the MCF design point $\mathbf{u}^*_{\text{MCF}}$. As WIS is defined in the physical space, the corresponding design point $\mathbf{x}^*$ is obtained by applying the inverse isoprobabilistic transformation: $\mathbf{x}^* = T^{-1}\left(\mathbf{u}^*_{\text{MCF}}\right)$. The instrumental pdf is then defined with normal distributions for which the means and standard deviations are given in Table A.5.

Table A.5: Instrumental pdf $H_{\mathbf{X}}$ for applying WIS on the penstock use-case.

X_i	Distribution	μ	σ
$X_1 = R_m$ (MPa)	Normal	$x^*_{R_m}$	σ_{R_m}
$X_2 = \varepsilon$ (MPa)	Normal	x^*_{ε}	σ_{ε}
$X_3 = \Delta e_{\text{corr}}$ (mm)	Normal	$x^*_{\Delta e_{\text{corr}}}$	$\sigma_{\Delta e_{\text{corr}}}$
$X_4 = \Delta e_{\text{extra}}$ (mm)	Normal	$x^*_{\Delta e_{\text{extra}}}$	$\sigma_{\Delta e_{\text{extra}}}$
$X_5 = a$ (mm)	Normal	x^*_a	$0.25x^*_a$
$X_6 = K_{\text{IC}}$ (MPa.$\sqrt{\text{m}}$)	Normal	$x^*_{K_{\text{IC}}}$	$0.25x^*_{K_{\text{IC}}}$

Appendix B. Artificial neural networks architectures

The first tuning parameters to be set when constructing ANNs are the number of hidden layers, the corresponding number of neurons and the activation functions to be used. There is no precise rule for assessing the right numbers of hidden layers and neurons. Generally, the higher the input dimension and the complexity of the response behavior, the more hidden layers and neurons are needed. Table B.6 presents the chosen architectures for ANN_1 and ANN_2 which are the same except for the output layer as the output of ANN_1 is a single classification probability ($p_1 \leq 0.5$ means that there is no root and $p_1 > 0.5$ means that are roots) and the output of ANN_2 corresponds to the two predicted roots. The activation functions are also the same with the use of "ReLU" except for the output layer where "sigmoid" is used for generating the classification probability and "linear" is used for the regression problem. Different architectures have not been tested as high performances of both ANNs were quickly achieved.

The next parameters to define are the ones directly involved for the training process, namely the loss function, the loss function optimizer, the metric used for validation and the number of epochs. The choices made in the present paper are given in Table B.7. Both loss functions "binary crossentropy" and "mean squared error" are the most considered ones for classification and regression problems respectively. The loss optimizer "Adam"

Table B.6: Architectures of ANN_1 and ANN_2.

Layer	Number of neurons		Activation function	
	ANN_1	ANN_2	ANN_1	ANN_2
Input layer	10	10	–	–
Hidden layer 1	64	64	ReLU	ReLU
Hidden layer 2	32	32	ReLU	ReLU
Hidden layer 3	16	16	ReLU	ReLU
Output layer	1	2	sigmoid	linear

Table B.7: Training parameters of ANN_1 and ANN_2.

Parameters	ANN_1	ANN_2
Loss function	binary crossentropy	mean squared error
Loss optimizer	Adam	Adam
Validation metric	accuracy	mean squared error
Epochs	50	50

is very common in deep learning and is known to converge efficiently. The validation metric is used to quantify the quality of the trained ANNs on the validation samples. It is very important as the trained ANNs that are saved are the ones that correspond to the epoch with the best validation metric. "Accuracy" (the proportion of correct classifications) and "mean squared error" are very common for classification and regression purposes respectively. The number of epochs plays an important role on the learning process. Too few epochs might lead to an underfit model which means that the training process did not enable the model to understand well all the features. Conversely, too many epochs might lead to an overfit model which means that it only performs well on the training inputs but not on new inputs. However, there are ways to circumvent this issue. In the present paper, a checkpoint is applied so that the model that is saved is the one that performs best on the validation data. Figures B.24 and B.25 present the convergence of the accuracy of ANN_1 and of the loss of ANN_2. The fact that the best configuration of ANN_2 is obtained at the last epoch suggests that more epochs might have improved the metric. However, both metrics are satisfactory.

Funding

The first author is involved in a Ph.D. program funded by EDF (CIFRE).

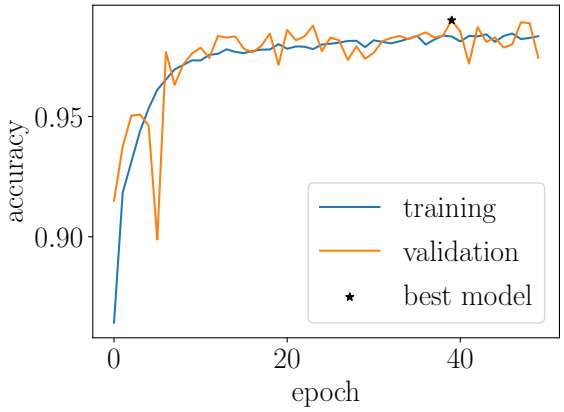


Figure B.24: Evolution of the accuracy of ANN_1 on the training and validation samples for $n_{\mathrm{train}} = 3 \times 10^4$.

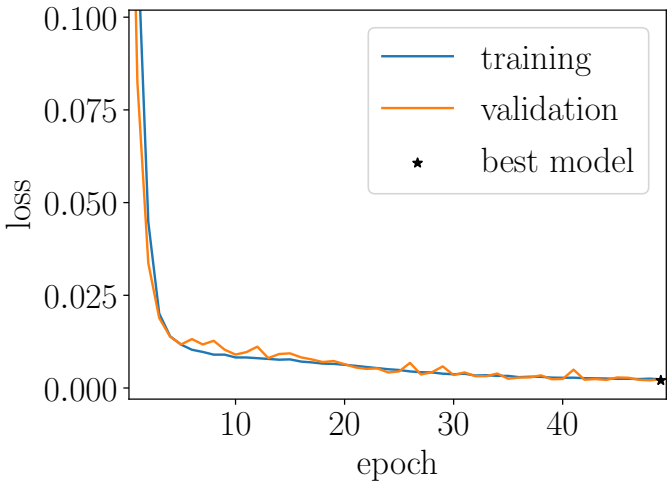


Figure B.25: Evolution of the loss of ANN_2 on the training and validation samples for $n_{\mathrm{train}} = 3 \times 10^4$.

References

[1] M. Lemaire, Structural Reliability, Wiley & Sons, 2009.

[2] E. Ardillon, et al., SRA into SRA: Structural reliability analyses into system risk assessment, an ESReDA collective book, Det Norske Veritas (2010).

[3] A. D. Kiureghian, Aleatory or epistemic? does it matter?, Structural Safety 31 (2009) 105–112.

[4] S. Göhler, T. Eifler, T. Howard, Robustness metrics: consolidating the multiple approaches to quantify robustness, Journal of Mechanical Design 138 (2016) 111407.

[5] Y. Ben-Haïm, Info-Gap Decision Theory: Decisions under Severe Uncertainty, Elsevier, 2006.

[6] I. Takewaki, Y. Ben-Haïm, Info-gap robust design with load and model uncertainties, Journal of Sound and Vibrations 288 (2005) 551–570.

[7] Y. Kanno, S. Fujita, Y. Ben-Haïm, Structural design for earthquake resilience: Info-gap management uncertainty, Structural Safety 69 (2017) 23–33.

[8] J. Hall, R. Lempert, K. Keller, A. Hackbarth, C. Mijere, D. McInerney, Robust climate policies under uncertainty: a comparison of robust decision making and info-gap methods, Risk Analysis 32 (2012) 1657–1672.

[9] E. Matrosov, A. Woods, J. Harou, Robust decision making and info-gap decision theory for water resource system planning, Journal of Hydrology 494 (2013) 43–58.

[10] F. Hemez, Y. Ben-Haïm, Info-gap robustness for the correlation of tests and simulations of a non-linear transient, Mechanical Systems and Signal Processing 18 (2004) 1443—1467.

[11] A. Ajenjo, E. Ardillon, V. Chabridon, B. Iooss, S. Cogan, E. Sadoulet-Reboul, An info-gap framework for robustness assessment of epistemic uncertainty models in hybrid structural reliability analysis, Structural Safety 96 (2022) 102196.

[12] Y. Ben-Haïm, Uncertainty, probability and information-gaps, Reliability Engineering & System Safety 85 (2004) 249—266.

[13] S. Ferson, W. T. Tucker, Probability boxes as info-gap models, in: Proceedings of the North American Fuzzy Information Processing Society, IEEE, New York City, 2008.

[14] J. Morio, M. Balesdent, Estimation of rare event probabilities in complex aerospace and other systems: a practical approach, Elsevier, 2015.

[15] P. Koutsourelakis, H. Pradlwarter, G. Schueller, Reliability of structures in high dimensions, part I: algorithms and application, Probabilistic Engineering Mechanics 19 (2004) 409–417.

[16] H. Pradlwarter, G. Schuëller, P. Koutsourelakis, D. Charmpis, Application of line sampling simulation method to reliability benchmark problems, Structural Safety 29 (2007) 208–221.

[17] E. Ardillon, P. Bryla, A. Dumas, Probalistic optimization of margins for plastic collapse in the mechanical integrity diagnoses of penstocks, in: Congrès Lambda Mu 21, Reims, 2018.

[18] E. Ardillon, P. Bryla, A. Dumas, Penstock reliability assessments: some results and developments, in: Proceedings of the 13th International Conference on Structural Safety and Reliability (ICOSSAR), Shanghai, 2021.

[19] M. Baudin, A. Dutfoy, B. Iooss, A.-L. Popelin, OpenTURNS: An industrial software for uncertainty quantification in simulation, in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer, 2017, pp. 2001–2038.

[20] L. Tvedt, Proban – probabilistic analysis, Structural Safety 28 (2005) 150–163.

[21] M. Valdebenito, P. Wei, J. Song, M. Beer, M. Broggi, Failure probability estimation of a class of series systems by multidomain Line Sampling, Reliability Engineering & System Safety 213 (2021) 107673.

[22] M. D. Angelis, E. Patelli, M. Beer, Advanced Line Sampling for efficient robust reliability analysis, Structural Safety 52 (2015) 170–182.

[23] G. Alefeld, F. Potra, Y. Shi, Algorithm 748: enclosing zeros of continuous functions, ACM Transactions on Mathematical Software 21 (1995) 327—344.

[24] Y. Ben-Haïm, I. Elishakoff, Convex models of uncertainty in applied mechanics, Elsevier, 1990.

[25] D. Finkel, Direct optimization algorithm user guide, Technical Report, Center for Research in Scientific Computation, North Carolina State University (2003).

[26] D. Jones, J. Martins, The DIRECT algorithm: 25 years Later, Journal of Global Optimization 79 (2021) 521—566.

[27] M. Faes, M. Daub, S. Marelli, E. Patelli, M. Beer, Engineering analysis with probability boxes: A review on computational methods, Structural Safety 93 (2021) 102092.

[28] R. Schöbi, B. Sudret, Structural reliability analysis for p-boxes using multi-level meta-models, Probabilistic Engineering Mechanics 48 (2017) 27–38.

[29] I. Depina, T. Le, G. Fenton, G. Eiksund, Reliability analysis with metamodel line sampling, Structural Safety 60 (2016) 1–15.

[30] X. Yuan, S. Liu, M. Valdebenito, J. Gu, M. Beer, Efficient procedure for failure probability function estimation in augmented space, Structural Safety 92 (2021) 102104.

[31] V. Chabridon, M. Balesdent, J.-M. Bourinet, J. Morio, N. Gayton, Evaluation of failure probability under parameter epistemic uncertainty: application to aerospace system reliability assessment, Aerospace Science and Technology 69 (2017) 526–537.

[32] X. Yuan, Z. Zhenxuan, Z. Baoqiang, Augmented line sampling for approximation of failure probability function in reliability-based analysis, Applied Mathematical Modelling 80 (2020) 895–910.

[33] X. Yuan, Local estimation of failure probability function by weighted approach, Probabilistic Engineering Mechanics 34 (2013) 1–11.

[34] A. Jain, J. Mao, K. Mohiuddin, Artificial neural networks: a tutorial, Computer 29 (1996) 31—44.

[35] R. Hecht-Nielsen, Theory of the backpropagation neural network, in: International Joint Conference on Neural Networks, IJCNN, 1989.

[36] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11—26.

[37] M. Papadrakakis, N. Lagaros, Reliability-based structural optimization using neural networks and Monte Carlo simulation, Computer Methods in Applied Mechanics and Engineering 191 (2002) 3491–3507.

[38] V. Papadopoulos, D. Giovanis, N. Lagaros, M. Papadrakakis, Accelerated subset simulation with neural networks for reliability analysis, Computer Methods in Applied Mechanics and Engineering 223 (2012) 70–80.

[39] E. Zio, N. Pedroni, An optimized Line Sampling method for the estimation of the failure probability of nuclear passive systems, Reliability Engineering & System Safety 95 (2010) 1300–1313.

[40] A. Chojaczyk, A. Teixeira, L. Neves, J. Cardoso, C. G. Soares, Review and application of Artificial Neural Netorks models in reliability analysis of steel structures, Structural Safety 52 (2015) 78–89.

[41] J. Song, P. Wei, M. Valdebenito, M. Beer, Active learning line sampling for rare event analysis, Mechanical Systems and Signal Processing 147 (2021) 107113.

[42] M. Valdebenito, H. Jensen, H. Hernández, L. Mehrez, Sensitivity estimation of failure probability applying line sampling, Reliability Engineering & System Safety 171 (2018) 99–111.