



HAL
open science

Knowledge-driven System Simulation for Scenario Analysis in Risk Assessment

Pietro Turati, Nicola Pedroni, Enrico Zio

► **To cite this version:**

Pietro Turati, Nicola Pedroni, Enrico Zio. Knowledge-driven System Simulation for Scenario Analysis in Risk Assessment. Terje Aven, Enrico Zio. Knowledge in Risk Assessment and Management, John Wiley & Sons, Ltd, pp.165-219, 2017, 9781119317890. 10.1002/9781119317906.ch8 . hal-01989144

HAL Id: hal-01989144

<https://hal.science/hal-01989144>

Submitted on 22 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/322039177>

Knowledge-driven System Simulation for Scenario Analysis in Risk Assessment

Chapter · January 2018

DOI: 10.1002/9781119317906.ch8

CITATIONS

0

READS

89

3 authors:



Pietro Turati

CentraleSupélec

12 PUBLICATIONS 37 CITATIONS

[SEE PROFILE](#)



Nicola Pedroni

Politecnico di Torino

108 PUBLICATIONS 749 CITATIONS

[SEE PROFILE](#)



Enrico Zio

Politecnico di Milano

887 PUBLICATIONS 12,092 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Advanced computational methods for modelling the mechanisms of degradation in equipments of electricity production plants and uncertainty modelling and propagation

[View project](#)



Neural Network Modeling for Prediction under Uncertainty in Energy System Applications [View project](#)

Knowledge-Driven System Simulation for Scenario Analysis in Risk Assessment

Turati Pietro, Pedroni Nicola, Zio Enrico

Acronyms

AEMO	: Australian Energy Market Operator	MC	: Monte Carlo
AK-MCS	: Adaptive Kriging-Monte Carlo Simulation	MCMC	: Markov Chain Monte Carlo
ANN	: Artificial Neural Network	M-H	: Metropolis-Hastings
AR	: Acceptance Ratio	MM	: Meta-Model
BIS	: Bank of International Settlement	MS	: Main Source
Cdf	: Cumulative density function	MVL	: Multiple-Value Logic
CR	: Critical Region	NFE	: Number of simulations need for the First complete Exploration
CSN	: Nuclear Security Council	NPP	: Nuclear Power Plant
DD	: Damage Domain	NSE	: Number of simulations need for the Second complete Exploration
DE	: Differential Evolution	NSS	: Not Supplied Set
DET	: Dynamic Event Tree	PCE	: Polynomial Chaos Expansion
DEX	: Deep EXploration	PCP	: Parallel Coordinate Plot
DOE	: Design Of Experiments	QMC	: Quasi Monte Carlo
ET	: Event Tree	RSM	: Response Surface Method
ENS	: Energy Not Served	SAMG	: Severe Accident Management Guidelines
ES	: End-State	SLOCA	: Seal Leak Of Coolant Accident
GOC	: Gas provided in Overloaded Conditions	SoS	: System of Systems
GSC	: Gas provided in Safe Conditions	SPLOM	: Scatter PLOt Matrix
I/O	: Input/Output	SVM	: Support Vector Machine
INL	: Idaho National Laboratories	UCR	: Unexplored Critical Region
ISA	: Integrated Safety Assessment	UECR	: Unexplored Extreme Critical Region
kNN	: k Near Neighbor		
LOF	: Local outlier Factor		
LOO	: Leave-One-Out		

1 Introduction

In recent times, discussions have arisen on the fundamental concept of “risk” and other foundational issues related to its assessment (Aven, 2012a, 2012b, 2016b; Cox, 2015). From a general perspective, it is understood that the outcomes of risk assessment are conditioned on the knowledge and information available on the system and/or process under analysis (Aven, 2016a; Aven & Zio, 2014; Zio, 2016b).

Recognizing this, leads to accepting the inevitable existence of a residual risk to be dealt with, related to the unknowns in the system and/or process characteristics and behaviors.

Then, it is important to be aware of the incomplete knowledge conditioning the assessment outcomes, somewhat along the lines of thought of the former United State Secretary of Defense, Donald Rumsfeld, who said the following at the press briefing on 12 February 2002, addressing the absence of evidence linking the government of Iraq with the supply of weapons of mass destruction to terrorist groups (Aven, 2013):

“There are known knowns: things we know we know. We also know there are known unknowns: that is to say, we know there are some things we do not know. But there are also unknown unknowns: the one we don’t know we don’t know.”

Correspondingly, different events can be classified according to the degree of knowledge available for the risk assessment (Flage & Aven, 2015):

1. Unknown-unknown
2. Unknown-known
3. Known-unknown
4. Known-known

In particular: 1) identifies those events that were unknown to everyone, at the time of the risk assessment; 2) indicates those events unknown to the risk analysts performing the assessment, but known to someone else; 3) identifies situations of awareness where the background knowledge is weak but there are indications or justified beliefs that a new, unknown type of event (new in the context of the activity) could occur in the future; 4) indicates events that are known to the analysts performing the risk assessment, and for which evidence exists.

According to (Flage & Aven, 2015), events and scenarios belonging to 1-2 and 4, and associated to negligible probabilities of occurrence, are black swans in the sense of (Taleb, 2007), whereas category 3 is representative of emerging risks, defined as new risks or familiar risks that become apparent in new or unfamiliar conditions (2015, International Risk Governance Council, IRGC). Note that, clearly, the concepts of “new” and “unfamiliar” are dependent on the background knowledge available.

For the sake of giving an example, consider the South Australia power network, which underwent a massive blackout caused by a cascading failure triggered by a heavy storm on the 28th Sep 2016. Around 1.7M people remained without power for 3h and some days were necessary to restore completely the energy supply. According to the preliminary report of the Australian Energy Market Operator (AEMO), the

heavy storm was a “non-credible event”, i.e., either an unknown-known or a known-known with a negligible probability associated (AEMO, 2016).

From the above qualitative discussion, we can retain that risk assessment amounts to a systematic and structured effort to present the knowledge and information available on events, processes and scenarios that affect specific decisions to be made for the management of risk. Risk assessment can be seen as a tool for organizing the knowledge that analysts have, on the system of interest (Flage & Aven, 2015).

When the unknowns and uncertainties in the assessment are many and the object of the assessment is a complex system, identifying and characterizing scenarios and conditions leading to critical situations becomes not trivial: a large set of scenarios and conditions is possible, and only few, rare ones are of interest because leading to critical situations.

In this chapter, we investigate the possibility of using system simulation for scenario analysis, to increase the knowledge on the response of a system to different conditions, with the aim of identifying possible unexpected or emergent critical state of the system. Indeed, verified and validated numerical models (or “simulators”) offer an opportunity to increase the knowledge regarding the system under analysis. Within a simulation-based scenario analysis, the analyst can run a number of simulations with different initial configurations of the system design and operation parameters, and identify a posteriori those leading to critical system states. These states form the so called “Critical Regions” (CRs) or “Damage Domains” (DDs) (Montero-Mayorga, Queral, & Gonzalez-Cadelo, 2014). The identified CRs can correspond to the prior knowledge of the analyst, i.e., the analyst is already aware that those configurations lead to critical outputs; or, be “surprising”, i.e., the analyst is not aware of such potential consequences and be “surprised” by them.

In the remainder of the chapter, we address the following issues with respect to the contribution of system simulation to risk assessment: *i*) challenges in simulation-based CR exploration (Section 2); *ii*) existing methods (Section 3); *iii*) two approaches proposed by the authors to drive scenarios exploration for CR identification (Section 4). Finally, in Section 5 some conclusions are drawn and future perspectives are discussed.

2 Problem statement

Simulation models of system behavior can be complex because:

- *High-dimensional*, i.e., with a large number of inputs and/or outputs;
- *Nonlinear*, due to the complexity of the relationships among the system elements;
- *Dynamic*, because the system evolves in time;

- *Computationally demanding*, as a consequence of the above characteristics and of the numerical methods employed.

The high dimensionality in the inputs implies that the conditions and scenarios to explore, and the corresponding system end-states to check for the identification of the CRs, increase exponentially with the space dimensions (Zio, 2014). Also, it challenges the effective visualization for interpretation of the results, calling for specifically designed representation tools. Similar issues arise also for the high dimensionality of the output space, where clustering techniques can be employed to identify groups of outputs having similar behavior, for their characterization as critical (Maio, Secchi, Vantini, & Zio, 2011; D. Mandelli et al., 2013; Mandelli, Yilmaz, Aldemir, Metzroth, & Denning, 2013).

Nonlinearities in the model usually make it difficult to predict which is the output associated to a specific input configuration, particularly in the inverse problem of interest of discovering the set of inputs leading the system to a specific (critical) output. In practice, when the computational model is a black box (because of empirical nature or because too complicated), the only feasible way to solve the problem is to run simulations and post-process the results to retrieve the information of interest from the generated data.

As for the analysis of dynamic systems, this calls for methods capable of dealing with (deterministic or stochastic) changes occurring during the time horizon of the analysis (by simulations), e.g., sequences of events occurring (possibly stochastically, e.g., components failures, or deterministically, e.g., due to control actions) at different times and that affect the operation of the system.

Under the conditions depicted above, typically encountered in practice, computational cost becomes an issue for simulation-based system response analysis for risk assessment. Indeed, the high computational cost for a single simulation prevents the analyst from running and exploring a large number of configurations, as instead necessary to gain knowledge on the system CRs. Then, there is a need of methods capable of extracting information on the system, resorting to a limited number of well-designed simulations. To achieve this goal, the methods should be capable of automatically understanding, during the simulation, which configurations are most promising to explore the system CRs.

3 State of the art

In the context of risk assessment, the combination of Event Trees (ETs) (diagrams representing the sequential logic of the system response to accident initiating events) and mathematical models of the system dynamics has been advocated as the way for determining the End-States (ESs) that can be reached by the system in accident scenarios and for deriving the corresponding causality relations among the events

occurring in the scenarios (Aldemir, 2013; Li, Kang, Mosleh, & Pan, 2011; Siu, 1994; Zio, 2014). Works on Dynamic Event Trees (DETs) (Cepin & Mavko, 2002; Cojazzi, 1996; Hakobyan et al., 2008; Hsueh & Mosleh, 1996; Kloos & Peschke, 2006; Labeau, Smidts, & Swaminathan, 2000) have highlighted that the end-states reached by a system as a result of an accident scenario do not depend only on the order of occurrence of the events in the sequence of the accident scenario, but also on the exact time at which these events occur and on their magnitude (Aldemir, 2013; Di Maio, Baronchelli, & Zio, 2015a; Di Maio, Vagnoli, & Zio, 2015; Garrett & Apostolakis, 1999; Li et al., 2011; Smidts & Devooght, 1992). However, exploring all dynamic sequences amounts to moving in a system state space of theoretically infinite dimension (because of the continuous time and magnitude variables). To address this issue, the majority of the methods available in the literature proceed to a discretization of the time and magnitude dimensions to reduce the state space size, and/or the pruning of branches associated to sequences having low probability of occurrence. However, these techniques may miss “rare” sequences of interest because leading to CR outcomes (Hakobyan et al., 2008; Rutt et al., 2006).

To tackle these issues, some authors have introduced an adaptive simulation framework to drive the exploration of scenarios (i.e., ET branches) towards those having more uncertain outcomes (Hu, Groen, & Mosleh, 2004; Turati, Pedroni, & Zio, 2015). In simple words, the event times and magnitudes worth to be explored are those that can generate scenarios with outcomes different from those already identified. If sequences with different times of occurrence and magnitudes of the same events lead to exactly the same scenario outcome, thoroughly exploring them does not add any additional information on the system CRs. On the other hand, if the same scenario can lead to several outcomes for different occurrence times and magnitudes of its events, it is worth running many simulations to discover the relations between the occurrence time and magnitude of the events and the scenario outcomes.

As mentioned earlier, a fundamental issue in risk assessment is the identification of the so-called CRs DDs, i.e., the input configurations that lead the system to safety-critical outcomes. In mathematical terms, given a deterministic Input/Output (I/O) model $\mathbf{Y} = f(\mathbf{X})$, where the inputs \mathbf{X} are uncertain and where the outputs \mathbf{Y} are realizations of simulations, the objective is to identify the set of inputs satisfying specific conditions for the output, e.g., those having output values above given safety-critical thresholds $\chi = \{\mathbf{x} \text{ s. t. } \mathbf{y} \geq \mathbf{Y}_{thres}\}$, which correspond to critical system state, i.e. belonging to a CR. To search for these conditions, one approach is the Design Of Experiments (DOE) (Fang, Li, & Sudjianto, 2005; Kuhnt & Steinberg, 2010; Santner, Williams, & Notz, 2003), whereby a set of input configurations is selected with a given logic to probe the input state space, the corresponding outputs are computed by simulation and those leading to safety-critical outputs are identified. Then, these available I/O data are post-processed, e.g., by

means of expert analysis or machine learning, to get insights on the CRs such as: causality relations between inputs and outputs, safety-oriented characteristics, shapes and number of the CRs, etc. For example, the Spanish Nuclear Safety Council has developed an Integrated Safety Assessment (ISA) methodology that has been recently used to verify whether the current Severe Accident Management Guidelines (SAMG) are properly defined for a Seal Loss Of Coolant Accident (SLOCA) (Queral et al., 2016). Authors exploited the expert knowledge to limit the input state space within a specific domain. The reduced domain has been probed by means of several simulations, whose results allow a repartition of the state space according to the different types of consequences reached by the nuclear plant during the accident (e.g., core uncover, fuel melting, vessel failure, etc.). For this, a substantial expert knowledge has been involved in the post-processing to have a physical interpretation of the events characterizing the accident scenario and of the impact of time on the occurrence of a failure and its recovery. Despite the large number of simulations performed, only a single accident scenario has been analyzed due to the high computational cost. In (Di Maio, Bandini, Zio, Alfonsi, & Rabiti, 2016) the authors, in collaboration with the U.S. Idaho National Laboratories (INL), make use of a surrogate model to reproduce the limit surface that separate the CRs from the safety regions during a station black-out in a Boiling Water Reactor simulated by means of the nuclear safety code RELAP5-3D (RELAP5-3D, 2005). Then, the identified CRs are projected on the subspace of the controllable variables and the most safe operation conditions are identified as those that are more distant from the CRs limit surface by means of a K-D Tree algorithm (Bentley, 1975).

The identification of CRs leads to the identification of prime implicants, as an extension of the concept of minimal cut sets in the ET Analysis. Prime implicants are defined as the minimal sets of process parameters values and components failure states that are sufficient to cause a failure of the dynamic system. In (Di Maio, Baronchelli, et al., 2015a; Di Maio, Baronchelli, & Zio, 2015b), the authors proposed two different frameworks for prime implicants identification, upon discretization of the input space by means of Multiple-Value Logic (MVL). In the first paper, the authors employed a Differential Evolution (DE) algorithm for the identification of the prime implicants, whereas in the second paper they resort to a visual interactive method that allows retrieving the values of the main features characterizing the prime implicants sequences.

In parallel to the use of simulation for CRs identification, but with a slightly different objective, techniques for the falsification of temporal properties have been proposed (Dreossi et al., 2015; Fainekos, Sankaranarayanan, Ueda, & Yazarel, 2012; Nghiem et al., 2010). Dynamic systems are designed to satisfy certain specifications: for example, the liquid level of a tank is controlled by automatic valves to remain between two threshold values; falsification looks for trajectories that lead the system out of the design

specifications, i.e., “falsifying” the expected system behavior. Whereas falsification techniques aim at showing that at least one trajectory not satisfying the design specifications exists, CRs identification methods aim at discovering and characterizing all trajectories that do not satisfy the design specifications.

Furthermore, nowadays systems are more and more interconnected (Systems of Systems SoS) and new behavior can emerge unexpectedly (emergent behavior) (Zio, 2016a, 2016b). In (Kernstine, 2012), a method called ARGUS is proposed for discovering emergent behavior in dynamic SoS. In particular, an iterative adaptive DOE is combined with parallel computing. The method takes the advantages of the available computing technologies (cloud computing and clusters), keeping the efficiency and flexibility of an adaptive DOE. The adaptive algorithm is used to select at each iteration a batch of candidate configurations to explore, while a cluster of processors is employed to run in parallel the simulations. However, since the method has been specifically designed for the exploration of a stochastic model, it loses its advantages when applied to a deterministic one. In addition, ARGUS makes use of polynomial harmonics to estimate the mean of the response function, which have been shown not to be efficient in high dimensionality.

Nuclear and financial industries have recently increased their attention to extreme yet possible scenarios (Authority, 2016; Commission, 2013). For example, the European Commission in response to the 2011 Fukushima nuclear accident, has requested to all state members to perform specific stress tests to assess the resilience of the nuclear power plants to several typologies of extreme events: earthquakes, floodings, terrorist attacks and aircraft collisions. Similarly, the Bank for International Settlement (BIS) requires financial institutions to perform some stress tests for assessing their capacity and robustness against extreme financial scenarios (Sorge, 2004). Stress tests allow analysts to collect information regarding system response. However, the response is evaluated only with respect to extreme scenarios: thus, stress tests do not allow to discover whether among the normal range of input values and scenarios, critical events can emerge.

When the computational cost becomes a constraint for the analysis, meta-models (or, equivalently, surrogate models) can represent a possible viable solution (Gorissen, Couckuyt, Demeester, Dhaene, & Crombecq, 2010). Meta-models usually resort to a set of input/output observations obtained from the real model to train a “surrogate” capable of reproducing the behavior of the real model at a lower computational cost. Once the meta-model has been validated (e.g., by means of its out-of-sample prediction accuracy), it can be used to replace the real model and to simulate the behavior of the system. Many types of meta-models are available, each one with characteristics that suit specific conditions. Among the large number of methods available in the literature (Simpson, Poplinski, Koch, & Allen, 2001; Wang & Shan, 2007), we

recall here just some of them that have been used in the context of risk assessment: i) Polynomial Chaos Expansion (PCE), which resorts to a particular basis of the probability space to represent the real-model input/output relation (see Appendix B for details) (Sudret, 2008); ii) Response Surface Method (RSM), where usually a low-order set of polynomials is used to fit the data observations available and the corresponding polynomial coefficients can be estimated by linear regression (Myers, Montgomery, & Anderson-Cook, 2016); nonetheless, the intrinsic linearity of the method makes it not suitable for nonlinear models; iii) Artificial Neural Networks (ANNs) (and all the associated evolutions), which resort to a large set of models (neurons) connected by means of nonlinear transformations (network) for reproducing any model behavior, including nonlinear (Cheng & Titterington, 1994; Haykin & Network, 2004); nevertheless, ANNs usually require a large number of input/output observations for their training; iv) Support Vector Machine (SVM), which is capable of reproducing nonlinear behaviors by mapping the inputs in a larger feature space; in practice, the meta-model is linear between the mapped features and the output, but can be nonlinear between the input and the output (Clarke, Griebisch, & Simpson, 2004); v) Kriging, which makes use of a Gaussian process to exactly interpolate the available input/output observations, allowing at the same time to have an estimate and an associated confidence interval of the response function for any input configuration (Clarke et al., 2004; Kleijnen, 2009; Rasmussen & Williams, 2006); Kriging is especially indicated for reproducing nonlinear models that present humps and regional behavior (see Appendix C for details).

Many researchers have been developing toolboxes and software that support sequential DOE, meta-models, iterative sampling, simulation, etc. Among the others, we report: DAKOTA (Eldred et al., 2014) from the Sandia National Laboratories, UQLab (Marelli & Sudret, 2014) from the ETH of Zurich, OpenCOSSAN (Patelli, Broggi, Angelis, & Beer, 2014) from the Institute for Risk and Uncertainty of the University of Liverpool, SUMO (Gorissen et al., 2010) from the Surrogate Modeling lab of Ghent, SCAIS (Queral et al., 2016) from the Spanish Nuclear Safety Council (CSN), RAVEN from the INL (Alfonsi et al., 2016) and OpenTURNS from a collaboration of academic institutions and industrial companies such as EDF, Airbus and Phimeca (Baudin, Dutfoy, Iooss, & Popelin, 2016). Those tools are continuously updated and have an open version in matlab (UQLab, OpenCossan, SUMO) or in a developer C++/python source code (DAKOTA, SCAIS, RAVEN, OpenTURNS); also a commercial version with an associated interface is available for all of them, except for SUMO, RAVEN and OpenTURNS.

It must be pointed out that these software are not specifically designed to address the research issues here stated concerning the exploration of scenarios. Rather, they are designed to render the state of the art of many statistical analysis methods accessible to industry and practitioners. In any case, they remain a

practical starting point for reducing programming time and speed up the design process of new methods for model exploration and knowledge retrieval.

To sum up, the issue of knowledge retrieval by simulation for scenario exploration in risk assessment of safety-critical systems has been treated by two main approaches:

- *massive simulation*, which exploits parallel and cloud computing advancements for increasing the number of simulations;
- *adaptive simulation*, which makes use of machine learning algorithms to extract information from the available simulations and to use this information to “drive” the simulations towards the states of interest for the analysis, thus limiting the number of computationally expensive calls to the simulation model.

Meta-modeling can be used in both approaches to further reduce the computational cost. In what follows, two recently proposed adaptive strategies are presented, showing the efficiency and the added value that this kind of analyses can bring to the analyst.

4 Proposed Approaches

Two exploration strategies proposed by the authors for increasing knowledge in a risk assessment context are presented in this section. Both the theory underneath the methods and some simple, but representative, applications are given.

The first strategy has been designed to explore accident scenarios that could occur within a given dynamic system. In particular, it allows probing the time dimension and assessing the impact that time has on the progression of accident scenarios (Section 4.1). The second strategy aims at identifying the CRs, i.e., those configurations of inputs and parameters values that lead a given system to a critical output. The strategy has been developed with the main objective of dealing with high-dimensional systems described by computationally-demanding models: for this reason, particular attention has been devoted to assess the capability of limiting the number of calls to the numerical model used to precisely characterize the CRs (Section 4.2).

4.1 EXPLORATION OF EXTREME AND UNEXPECTED EVENTS IN DYNAMIC ENGINEERED SYSTEMS

4.1.1 Method

Accident scenario analysis requires to identify, list and analyze all possible failure scenarios that can occur to the system under analysis. DETs have been used to identify (dynamic) accident scenarios and characterize

their consequences. A large effort is required to consider the time dimension and its impact on the accident consequences. To keep the analysis feasible, methods have been introduced to either a priori discretize the time dimension and/or to prune some branches in the accident evolution. However, excluding branches having low probability of occurrence without considering the associated consequences and time discretization can miss possible "rare" critical accident sequences (Di Maio, Baronchelli, et al., 2015a; Garrett & Apostolakis, 1999; Li et al., 2011).

Before introducing the main characteristics of the method, some definitions should be given. We define a scenario as an ordered sequence of events in the life evolution of the dynamic system (i.e., within its mission time T_{Miss}), which may involve a particular group of components, safety functions or actions (e.g., mechanical failures, activation of safety systems and human decisions). For example, scenario S_1 could be defined by event A (failure of a component) at time T_A , followed by event B (failure of the safety system) at time $T_A < T_B < T_{Miss}$; scenario S_2 could be defined by the opposite order of the events B and A , with $T_B < T_A < T_{Miss}$. Since the events in the sequences may occur with the *same* order but at *different* times, an *infinite* number of sequences exist for a *single given* scenario, potentially leading to different outputs (i.e., system states), as demonstrated in (Di Maio, Baronchelli, et al., 2015a; Di Maio, Vagnoli, et al., 2015).

In accident progression analysis, which is the case in this section, the system output \mathbf{Y} usually represents the worst condition reached by the system during the simulation (Queral et al., 2016). In what follows, we define End-State (ES) a categorical variable synthetically representing the state of the system on the basis of its outputs. This is often the case in many applications. For example, in a Nuclear Power Plant (NPP) Loss-Of-Coolant Accident (LOCA) the output can be classified according to the different ESs reached by the reactor: core uncover, embrittlement condition, fuel melting, fuel relocation, vessel failure, etc., which correspond to consequences of different severity (Ibáñez et al., 2016).

The idea underlying the proposed strategy is that not all scenarios need to be explored with the same level of details. Indeed, consider two scenarios: one representing normal operation conditions, where no failures occur and one characterized by the occurrence of a component failure at time T_F and the corresponding repair at time T_R . Obviously, there is no interest in running many simulations exploring the normal condition scenario, since we already know its corresponding ES. In the component failure scenario, instead, we are interested in exploring the impact on the ES of the failure occurring at different times. Indeed, we can expect that if the repair is performed just after the failure, the impact of the component failure is lower than if it is performed later in the scenario.

For an efficient exploration of the scenarios, an adaptive simulation framework has been proposed by the authors (see Figure 1) (Turati, Pedroni, & Zio, 2016a). The framework is based on three main steps:

1) *preliminary exploration* (Section 4.1.1.1), i.e., a global exploration of the whole space of the dynamic system scenarios;

2) *interactive decision making* (Section 4.1.1.2), i.e., after the preliminary exploration, the analyst can decide to either improve his/her global view of the state space by increasing the number of simulations in the *preliminary exploration* (step 1), or focus the attention on a specific event of interest (step 3);

3) *deep exploration* (Section 4.1.1.3), i.e., a thorough exploration of a particular event: for example, the objective can be that of retrieving the possible evolutions within a specific scenario S_j that can potentially reach a given ES ES_i , indicated hereafter as the pair $\{S_j, ES_i\}$.

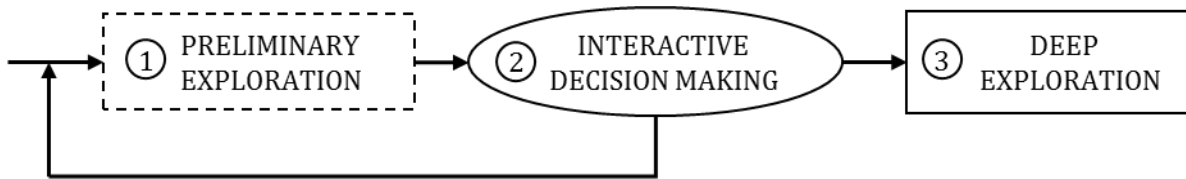


Figure 1 Sketch of the adaptive exploration framework

For generating time sequences within a scenario of interest, we resort to a joint uniform distribution over each scenario support (the region of variability of the times of occurrence of the ordered events in the scenario) in order to thoroughly explore the scenario and discover the whole set of possible ESs that each scenario can reach. To this aim, a Markov Chain Monte Carlo (MCMC) Gibbs sampling is employed (Robert & Casella, 2004).

4.1.1.1 Preliminary Exploration

Hereafter, we assume that preliminary exploration is run under the constraint of limited computational resources, i.e., of a fixed number of simulations to run. This step aims at enhancing the global knowledge regarding system dynamic behavior during accident scenarios. The exploration consists of two steps: i) selection of the scenario to explore according to a driving function; ii) simulation of a time sequence within the selected scenario.

The driving function should be flexible enough to take into account different analyst objectives and backgrounds. For example, the analyst could be interested in exploring and collecting information regarding the scenarios leading to a specific set of ES ES^* , e.g., the most critical ones. In this light, the choice of the

scenario during the preliminary exploration is made by selecting the scenario S^* which maximizes the driving function $I_{\gamma,\beta}(S_j, ES^*)$:

$$S^* = \operatorname{argmax}_{j \in \mathcal{S}} I_{\gamma,\beta}(S_j, ES^*), \quad (1)$$

where $I_{\gamma,\beta}(S_j, ES^*)$ is defined as:

$$I_{\gamma,\beta}(S_j, ES^*) = I_{\gamma,\beta}(N_j^{ES}, n_j, I_{ES^*}) = \begin{cases} \frac{(N_j^{ES})^\gamma}{n_j}, & I_{ES^*} = 0 \\ \frac{(N_j^{ES})^\gamma}{n_j} \cdot \beta, & I_{ES^*} = 1 \end{cases}, \quad (2)$$

where N_j^{ES} is the number of ESs that Scenario S_j can reach (if this information is not available, then it represents the number of ESs that have already been visited within the scenario and it is updated whenever a new ES is discovered by a new simulation run); n_j is the number of simulations that have already been run within S_j ; I_{ES^*} is a Boolean variable, which equals 1 if the simulations of scenario S_j can reach at least one of the ESs in ES^* , and 0 otherwise; $\gamma \in (-\infty, +\infty)$ and $\beta \in (1, +\infty)$ are two design parameters which reflects the preference of the analyst: γ represents analyst preference concerning scenario variability, whereas β represents analyst preference concerning an ES set ES^* . If $\gamma < 0$, the driving function chooses more frequently those scenarios that can reach a small number of ESs; if $\gamma = 0$, no preference is given to any scenario on the basis of its variability; otherwise, if $\gamma > 0$, the driving function selects more likely those scenarios that can reach a large number of ESs. Meanwhile, the higher β value, the more frequently the algorithm selects those scenarios that can reach an ES belonging to ES^* . It is worth noting, that if $\beta = 1$, no preference are given to any ES.

For the sake of clarity, two examples are here reported to separately show the impact of the two preference parameters. Consider a simple dynamic system where only four scenarios can occur S_1, \dots, S_4 and where each scenario can reach a different number of ESs, $N_1^{ES} = 1, N_2^{ES} = 2, \dots, N_4^{ES} = 4$. Finally, let us assume that all reachable ESs in the same scenario have the same probability of occurring and that the analyst has no preference regarding the ES to explore, i.e., $\beta = 1$. Table I reports the average of 1000 explorations, performed with 100 simulations each, that have been distributed among the different scenarios according to three different values of the parameter γ , i.e., $\gamma = -1$ (left), $\gamma = 0$ (middle) and $\gamma = 1$ (right).

Table I Average results of 1000 experiments. Each experiment runs 100 simulations of preliminary exploration with different values of the parameter γ : -1 (left); 0 (middle); 1 (right) and $\beta = 1$. Column "Tot" represents the total number of simulations run within the respective scenario.

	ES_1	ES_2	ES_3	ES_4	Tot	ES_1	ES_2	ES_3	ES_4	Tot	ES_1	ES_2	ES_3	ES_4	Tot
S_1	47.9	0.0	0.0	0.0	47.9	25.0	0.0	0.0	0.0	25.0	10.0	0.0	0.0	0.0	10.0
S_2	12.0	11.9	0.0	0.0	23.9	12.4	12.6	0.0	0.0	25.0	10.0	9.9	0.0	0.0	20.0
S_3	5.3	5.3	5.3	0.0	15.9	8.4	8.3	8.3	0.0	25.0	10.0	10.0	10.0	0.0	30.0
S_4	3.0	3.1	3.0	3.1	12.2	6.2	6.2	6.3	6.3	25.0	10.0	10.0	9.9	10.1	40.0

The choice of parameter $\gamma = 1$ is particularly suitable because, in this case, the exploration algorithm distributes the simulations among all the scenarios in order to guarantee that each scenario S_j “gathers” a number of simulations proportional to the number N_j^{ES} of ESs that each scenario can “generate”.

Assuming now, instead, that the analyst has interest in gathering information about the most variable scenarios, i.e., $\gamma = 1$, and the most critical ESs, e.g., $ES^* = \{ES_3; ES_4\}$. Table II reports the effects of different choices of parameter $\beta = \{1; 2; 4\}$ on the final distribution of the simulation runs among the scenarios. If $\beta = 1$, the algorithm turns to the preliminary guided exploration described above (left); otherwise, if $\beta > 1$, the scenarios that can reach the set ES^* are favored in the selection step (middle, right).

Table II Average results of 1000 experiments. Each experiment run 100 simulations of preliminary exploration with $\gamma = 1$ and with different values of the parameter β : 1 (left); 2 (middle); 4 (right). Column “Tot” represents the total number of simulations run within the respective scenario.

	ES_1	ES_2	ES_3	ES_4	Tot	ES_1	ES_2	ES_3	ES_4	Tot	ES_1	ES_2	ES_3	ES_4	Tot
S_1	10.0	0.0	0.0	0.0	10.0	7.0	0.0	0.0	0.0	7.0	3.1	0.0	0.0	0.0	3.1
S_2	10.0	9.9	0.0	0.0	20.0	7.0	6.0	0.0	0.0	12.9	3.0	2.9	0.0	0.0	5.9
S_3	10.0	10.0	10.0	0.0	30.0	12.5	11.7	11.7	0.0	36.0	12.6	12.3	12.5	0.0	37.4
S_4	10.0	10.0	9.9	10.1	40.0	12.7	11.8	11.8	11.7	48.1	13.4	13.4	13.4	13.3	53.5

For the preliminary exploration, we have proposed only one function based on two parameters, which can reflect the analyst interest about scenario variability and a set of known ESs; however, a variety of functions could be used at this stage to drive the selection of scenarios according to other desirable criteria.

4.1.1.2 Interactive Decision Making

Every time a preliminary exploration is performed, matrices, such those reported in Table I and Table II, become available. Hence, based on the events visited (i.e., on the pairs Scenario-ES (S_j, ES_i)) and on the number of simulations that have been run to visit them, the analyst can decide either to increase the number of simulations according to the criteria adopted in the preliminary exploration phase or to perform a deeper and more refined exploration of specific events of interest. According to his/her preference, the

analyst has to iteratively choose the maximum allowable number of simulations that can be run according to the *preliminary* or *deep* exploration, respectively. In many cases, the dimension of the system (state space) and the variability of its behavior (in practice, the number of ESs a scenario can reach and the corresponding probabilities), are not known a priori; on the contrary, the computational cost needed for a system simulation can be known (e.g., in terms of average time per simulation). Then, the computational effort can be considered as a constraint that the analyst needs to take into account in accordance with his/her preferences among the different exploration criteria. In this respect, it must be noticed that the proposed method does not guarantee that the whole event space is probed: inevitably, if the computational capacity available (in practice, the total number of simulations that can be run) is small compared to the size of the system state space, only a limited number of ESs can be explored for each scenario.

4.1.1.3 Deep Exploration

The objective of the deep exploration is to identify as precisely as possible, which system evolutions (i.e., which transition times) can lead to a given event of interest. For the sake of clarity, we assume that an event of interest is defined as the pair (Scenario, ES) = (S_j, ES^*) ; nonetheless, with no loss of generality ES^* can represent also a set of ESs. Given the structure of the mathematical model, the guiding idea of the deep exploration is to generate time sequences “around” those that have already reached the event (S_j, ES^*) . In order to achieve this goal, we resort to a MCMC method, which allows to generate a set of random samples from any desired (namely, *target*) probability distribution p (Robert & Casella, 2004). In detail, we utilize a Metropolis-Hastings (M-H) algorithm (Chib & Greenberg, 1995) to sample components transition times uniformly on the support SES^* of the event of interest (S_j, ES^*) , in other words, to sample uniformly among the transition times that lead to the event of interest. The M-H algorithm consists of two steps: *i*) proposition of a new candidate \mathbf{T}^* (in this case, a vector of transition times) in accordance to a *proposal* distribution q ; *ii*) acceptance or rejection of the proposed time vector. The interested reader is referred to the Appendix A for more details on the algorithm.

Nevertheless, it must be underlined that the Acceptance Ratio (AR) between the proposed samples and the accepted ones plays a fundamental role. High acceptance ratios ($AR > 0.9$) are a symptom of a proposal q with too small variability, i.e., most of the proposed \mathbf{T}^* are too close to the original ones and, thus, the algorithm results too slow in probing the support SES^* ; on the contrary, small acceptance ratios ($AR < 0.2$) are a symptom of a proposal q with too high variability, i.e., most of the proposed \mathbf{T}^* are likely to fall out of the support of interest SES^* . In this respect, adaptive MCMC methods exploiting an adaptive proposal

distribution have been presented in the literature and can be employed at this stage to “optimally” fill the support SES^* of interest (Andrieu & Thoms, 2008; Roberts & Rosenthal, 2009).

Regarding the approach used to choose the number of simulations to run for performing the deep exploration, two criteria are proposed: (i) fixed number of simulations (as in the *preliminary exploration*, Section 4.1.1.1); (ii) level of filling of the support of the event of interest. For what concerns the second criterion, the idea is to keep on generating new simulation outcomes until SES^* is filled by an amount of points (i.e., configurations) that “sufficiently” cover the entire outcome variability. In detail, after the preliminary exploration a set of occurrence time vectors $EX_V(SES^*) = \{\mathbf{T}_1, \dots, \mathbf{T}_V\}$ that lead to the event of interest (S_j, ES^*) is available. As a measure of the (time) space filling, the maximum of the minimum distances among these time vectors is considered: then, a time filling index $D_V(EX_V(SES^*))$ after the preliminary exploration is computed as:

$$D_V(EX_V(SES^*)) = \max_{i \in EX_V(SES^*)} \min_{j \neq i} d(\mathbf{T}_i, \mathbf{T}_j) \quad (3)$$

where $d(\cdot, \cdot)$ represents a proper distance between two vectors. Herein, for example, we consider the Euclidean one. Whenever a new time vector \mathbf{T}_n is accepted during the exploration, it is added to the set of time vectors that lead to the event of interest, i.e., $EX_n(SES^*) = \{EX_{n-1}(SES^*); \mathbf{T}_n\}$, and the filling index $D_n(EX_n(SES^*))$ is consequently updated. The deep exploration ends when the ratio between the current filling index and the preliminary one falls below a fixed threshold $\delta \in [0, 1]$, i.e., when the “density” of time vectors in the support SES^* of interest is $\sim(1/\delta)^l$ times higher than the preliminary one, being l the size of the time vector \mathbf{T}_n . Thus, the space filling capability of the algorithm is strictly related to the dimension of the vectors involved: in practice, the higher the dimension, the larger the number of random vectors needed to reduce the filling index. In this light, a maximum allowable number n_{max} of samples is also set, in order to limit in any case the maximum computational effort. Then, the stopping criterion becomes:

$$\frac{D_n(EX_n(SES^*))}{D_V(EX_V(SES^*))} < \delta \text{ or } n > n_{max} . \quad (4)$$

The corresponding algorithm is summarized in Table III.

Table III Sketch of the algorithm describing the deep exploration stopping criterion.

1. For $i = 1, \dots, V$ evaluate the minimum distances from the vector \mathbf{T}_i and save them in the vector \mathbf{d}_V :
$d_V(i) = \min_{j \neq i} d(\mathbf{T}_i, \mathbf{T}_j)$.
According to this notation $D_V(EX_V(SES^*)) = \max \mathbf{d}_V$.
2. Given a new time vector \mathbf{T}_n , update the \mathbf{d}_{n-1} vector for $i = 1, \dots, n - 1$:

$$d_n(i) = \min(d_{n-1}(i), d(\mathbf{T}_i, \mathbf{T}_n)),$$

3. Add the n -th component to \mathbf{d}_{n-1} resorting to the distance already available from the previous step:

$$d_n(n) = \min_{j \neq n} d(\mathbf{T}_n, \mathbf{T}_j).$$

4. Evaluate the filling index:

$$D_n(EX_n(SES^*)) = \max \mathbf{d}_n.$$

5. Check if the stopping criteria are satisfied:

$$\frac{D_n(EX_n(SES^*))}{D_V(EX_V(SES^*))} < \delta \text{ or } n > n_{max}$$

If not, return to step 2.

4.1.2 Gas Transmission Subnetwork

The case study under analysis is a gas transmission subnetwork composed of two pipes in parallel and another one in series. The input of each pipe is controlled by a valve. The block diagram is shown in Figure 2, where each pair valve-pipe is considered as a single block.

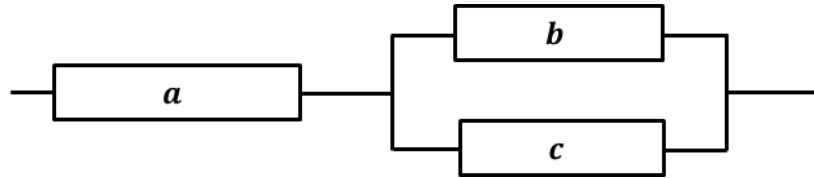


Figure 2 Block diagram of the system under analysis.

Each pipe can transmit gas with a maximum flow rate of $[\phi_a, \phi_b, \phi_c] = [8,5,5] \cdot 10^4 \text{ m}^3/\text{day}$, for pipes a, b, c , respectively. A control system adjusts the opening of the valves in order to guarantee the equilibrium between the input and output flows. Figure 3 shows the ET containing all the scenarios that can occur in the system. If one of the pipes in parallel breaks, the control system immediately closes the corresponding valve and increases the flow rate of the remaining pipe to the maximum, in order to compensate for the diminished flow. No reparation strategies are considered. The system presents 8 possible scenarios with different operating conditions: *i) safe*, i.e., all pipes are functioning correctly; *ii) overloaded*, i.e., one of the pipes in parallel is closed; *iii) broken*, i.e., no gas is provided by the system.

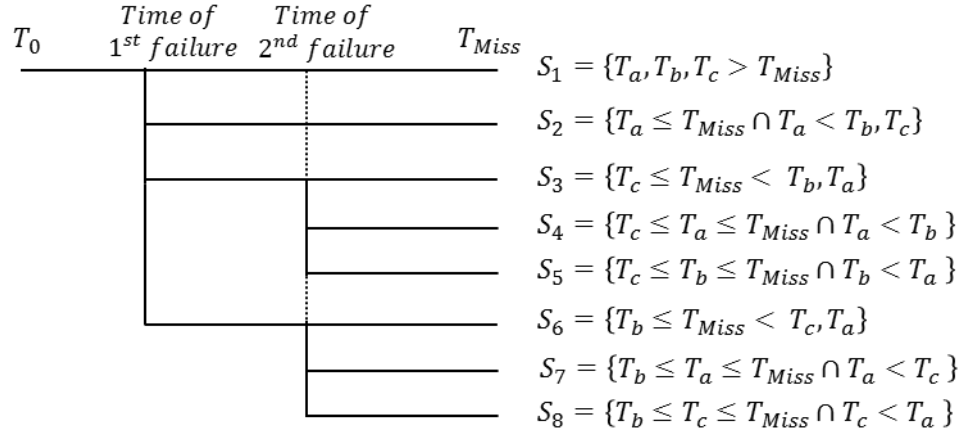


Figure 3 Event tree representation of the 8 scenarios that can occur, where T_a, T_b, T_c are the times of failures of components a, b, c , respectively, and T_{Miss} is the mission time.

The ESs for each scenario have been defined and classified on the basis of two output variables Y_1, Y_2 : *i*) the amount of Gas provided in Safe Conditions ($GSC = Y_1$), i.e., when all the components are functioning correctly; *ii*) the amount of Gas provided in Overloaded Conditions ($GOC = Y_2$), i.e., when one of the two pipes in parallel is down and the remaining one works at its maximum flow rate. With respect to that, GSC_{max} and GOC_{max} indicate the maximum quantities of gas that can be provided within the mission time $T_{Miss} = 900d$, in safe and overloaded conditions, respectively, i.e., $GSC_{max} = \phi_a \cdot T_{Miss}$ and $GOC_{max} = \max(\phi_b, \phi_c) \cdot T_{Miss}$. The outputs are, then, divided into six ESs according to the criteria reported in Figure 4. For example, $ES_4 = \left\{ \frac{1}{3} GSC_{max} < GSC \leq \frac{2}{3} GSC_{max} \cap 0 \leq GOC \leq \frac{1}{3} GOC_{max} \right\}$, which means that the system has operated for a medium period of time in *safe* conditions ($\frac{1}{3} GSC_{max} < GSC \leq \frac{2}{3} GSC_{max}$) and, then, once it goes in *overloaded* conditions, it breaks down ($0 \leq GOC \leq \frac{1}{3} GOC_{max}$).

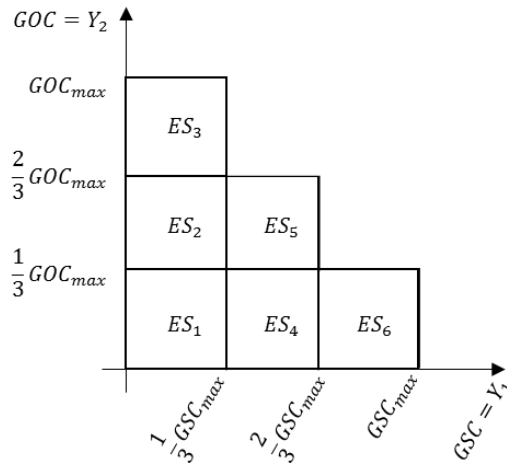


Figure 4 Classification of the ESs according to the 2 output variables GSC and GOC.

It must be noticed that not all the ESs can be reached by all scenarios. Table IV (left matrix) reports those ESs that can be reached by a given scenario (indicated by 1) and those that cannot (indicated by 0): each column in the Table represents an ES and each row represents a scenario. This information is usually not available a priori and, in general, its retrieval represents one of the objectives of the state space exploration. However, it is used here to analyze the performance of the proposed method. In Table IV (middle and right), two additional matrices show the reachable ESs for two sets of different gas flow rates, e.g., $[\phi_a, \phi_b, \phi_c] = [8, 3.7, 5] \cdot 10^4 \text{ m}^3/d$ and $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4 \text{ m}^3/d$, respectively. These values have been chosen in order to analyze the performance of the method for different parameters values, which imply that the number of reachable ESs varies.

Table IV Matrices of the end-states that the system can reach for each scenario for different sets of flow rate parameters values: $[\phi_a, \phi_b, \phi_c] = [8, 5, 5] \cdot 10^4 \text{ m}^3/d$ (left); $[\phi_a, \phi_b, \phi_c] = [8, 3.7, 5] \cdot 10^4 \text{ m}^3/d$ (middle) and $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4 \text{ m}^3/d$ (right).

	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6
S_1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1
S_2	1	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1
S_3	0	0	1	0	1	1	0	1	1	1	1	1	0	1	0	1	1	1
S_4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
S_5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1
S_6	0	0	1	0	1	1	0	0	1	0	1	1	0	0	1	0	1	1
S_7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S_8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

4.1.2.1 Preliminary Exploration

To evaluate the performance of the *preliminary exploration*, two indices are introduced: (i) the Number of simulations needed for the First complete Exploration (NFE), i.e., the number of simulations that should be run to visit at least once all the reachable ESs for all the scenarios; (ii) the Number of simulations needed for the Second complete Exploration (NSE), i.e., the number of simulations that should be run to visit all the reachable ESs for all the scenarios at least twice. NFE gives information about the number of simulations needed to explore all the events defined by the pairs (Scenario, ES) = (S, ES) , when the matrices shown in Table IV (i.e., the ESs) are not known yet. On the contrary, NSE gives information about how the simulations are efficiently distributed among the different scenarios, once the matrices in Table IV (i.e., the ESs) begin to be known as a result of the preliminary exploration. We analyzed two different situations: in the former, the analyst has a very poor background knowledge regarding the system, while in the latter he/she already

knows the system and is interested in collecting information regarding the scenarios that can reach a specific ES. For this reason, in the first case $\beta = 0$ and $\gamma = 1$; whereas in the second case, $\beta > 1$.

Considering the case with low prior knowledge, the results of the preliminary explorations are compared to those of: 1) a crude Monte Carlo simulation method (MC), that randomly selects the scenario and, then, simulates the proper transition times according to the same uniform sampling criterion proposed in Section 4.1.1; 2) an entropy-driven exploration (Turati et al., 2015), which follows a procedure similar to the *preliminary exploration*, but with an entropy-driven function instead of $I_{\gamma, \beta}(\cdot)$.

For all the gas flow rate reported in Table IV, the preliminary exploration has been performed 1000 times and the corresponding empirical cumulative density functions (cdfs) of NFE (left) and NSE (right) computed. Preliminary exploration achieve better or at least comparable performance than the entropy-driven exploration in all flow configurations tested. This is depicted in Figure 5-Figure 6 where the cdfs associated to the preliminary exploration (light-dashed line) are “shifted” to the left with respect to those associated to the entropy-driven exploration (dark-dotted line). On the other side, both the preliminary and the entropy-driven explorations largely outperform the MC one (light line) regarding both NFE and NSE. In particular, the difference is even larger in NSE, i.e., when the exploration algorithm is already aware of all the events (S, ES) that can occur. The results of flow configuration $[\phi_a, \phi_b, \phi_c] = [8, 3.7, 5] \cdot 10^4 \text{ m}^3/d$ are not depicted, due to the similarity with those in Figure 5. Finally, it must be noted that in one case the MC exploration is more effective than the other techniques (Figure 6, NFE). This is due to the fact that the rarest event (S, ES) occurs in a scenario that can reach a few number of end-states. However, while the entropy-driven method is stuck, the *preliminary* exploration allows changing parameter γ in order to increase the exploration effectiveness.

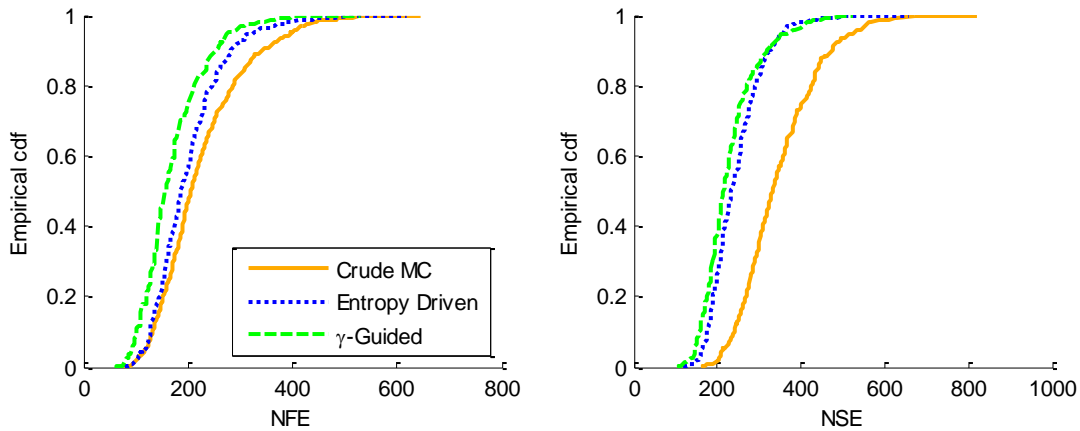


Figure 5 Empirical cdfs of the NFE (left) and of the NSE (right) for crude MC (light line), for an entropy-driven method (dark dotted line) and for the preliminary guided exploration with $\gamma = 1$ (light dashed line) with flow rate parameters $[\phi_a, \phi_b, \phi_c] = [8, 5, 5] \cdot 10^4 \text{ m}^3/d$.

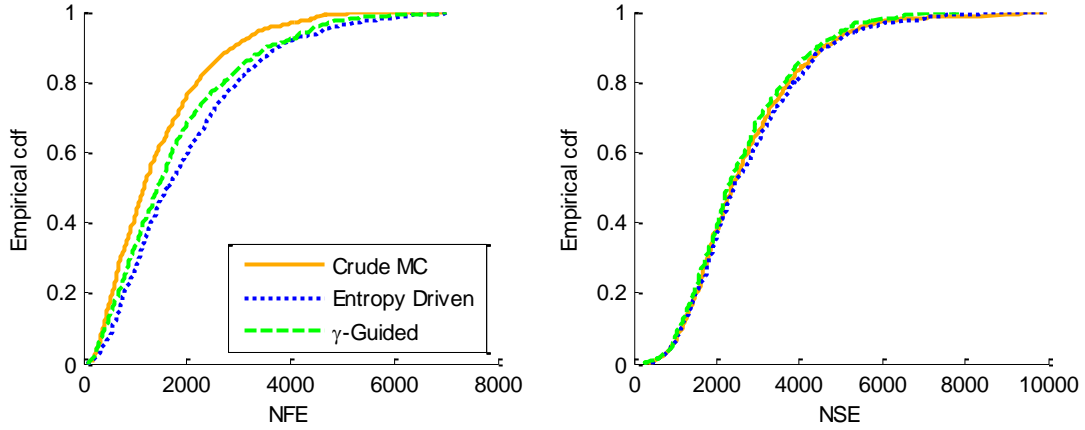


Figure 6 Empirical cdfs of the NFE (left) and of the NSE (right) for crude MC (light line), for an entropy-driven method (dark dotted line) and for the preliminary guided exploration with $\gamma = 1$ (light dashed line) with flow rate parameters $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4 \text{ m}^3/d$.

Considering now the case where the analyst has some prior knowledge, we consider the flow rate configuration $[\phi_a, \phi_b, \phi_c] = [8, 2.2, 6] \cdot 10^4 \text{ m}^3/d$ and we suppose that the analyst is interested in scenarios leading to ES_3 . To assess the impact of parameter β on the performance of the preliminary exploration, the average percentage increment of simulation falling into the scenarios of interest with respect to those falling in the same scenarios when no preference are given, i.e., $\beta = 1$, is computed for different values of $\beta = (2, 4, 8)$ and for different numbers of simulation runs $N_{simul} = [250; 500; 1000; 2000; 4000]$. 1000 experiments have been done for each combination of β and N_{simul} . Since similar behaviors have been observed for all scenarios leading to the ES of interest, only the boxplots associated to scenario S_7 are depicted in Figure 7. The larger the β value is, the larger the percentage increment, e.g., around (35, 60, 80)% for $\beta = \{2; 4; 8\}$, respectively. However, it must be noted that, if β is too large with respect to N_{simul} (e.g., $\beta = 8$ and $N_{simul} < 1000$), there is a high uncertainty in the performance. Indeed, if β is too large, the algorithm focuses its exploration effort, i.e., its simulation runs, on the first scenario that reach to the ES of interest, “preventing” the algorithm to discover other scenarios that can lead to the ES of interest. In particular, the larger the number of scenarios that can reach the ES of interest, the larger the sensitivity to the number of simulations, given β .

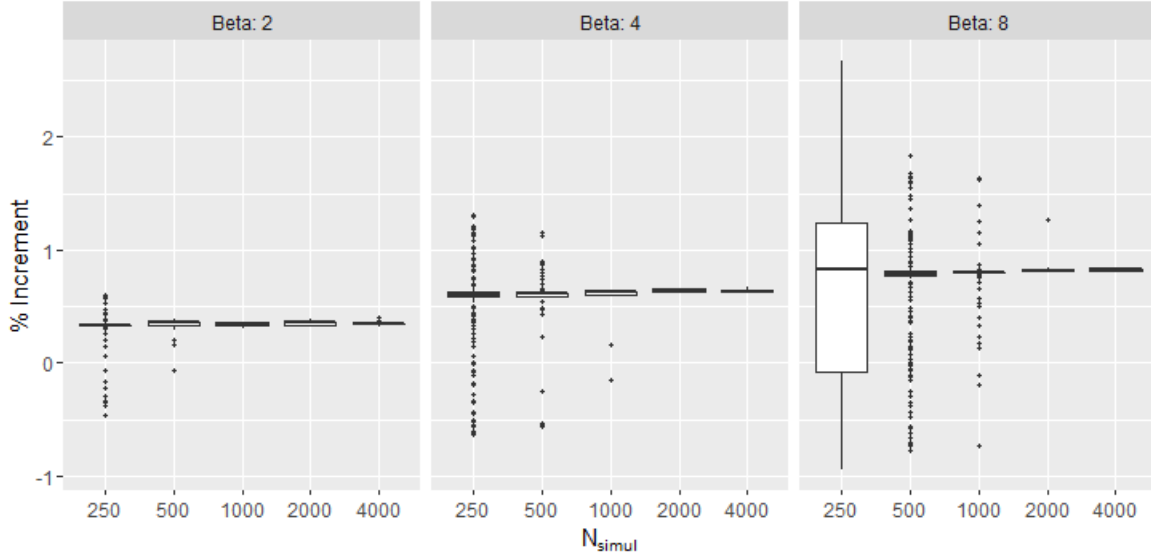


Figure 7 Boxplots of the percentage increment of simulations in a given scenario of interest S_7 , for parameter $\beta = \{2; 4; 8\}$ and for different numbers of simulations.

4.1.2.2 Deep Exploration

After a preliminary guided exploration of the system defined by parameters $[\phi_a, \phi_b, \phi_c] = [8, 3.6, 5] \cdot 10^4 \text{ m}^3/\text{d}$, a large variability in the outcomes is observed within scenario S_5 , as highlighted in Table V. Thus, it is interesting to retrieve the event time sequences that lead to two chosen ESs: ES_1 , which represents the worst final condition, and ES_3 , which has been visited only few times during the preliminary exploration.

Table V Matrix reporting the ESs visited by a preliminary guided exploration of the system with parameters $[\phi_a, \phi_b, \phi_c] = [8, 3.6, 5] \cdot 10^4 \text{ m}^3/\text{day}$, given a computational effort of 1000 simulations.

	ES_1	ES_2	ES_3	ES_4	ES_5	ES_6
S_1	0	0	0	0	0	29
S_2	21	0	0	38	0	28
S_3	0	27	10	24	36	47
S_4	46	29	0	41	5	23
S_5	39	50	2	57	7	18
S_6	0	0	23	0	28	36
S_7	38	36	22	36	14	26
S_8	34	39	24	41	12	22

The space filling parameter is set to 0.2 with a maximum number of simulations to run set to 5000. Multivariate Gaussian distributions have been used as proposal within the M-H algorithm. The covariance matrix associated to ES_1 has been estimated from the vectors of transient times obtained from the preliminary exploration. On the contrary, since only two vectors are available for ES_3 , a diagonal covariance

matrix with standard deviation equal to the Euclidean distance between the two vectors is considered. The chosen standard deviation provides an idea of the dimension of the support to explore. Figure 8 reports the transition time vectors of the scenario of interest S_5 after the preliminary exploration (on the left) and after the deep exploration (on the right). Results confirm that the proposed deep exploration is capable of increasing the number of simulations around the time sequences that reach the ES of interest. The results increase the knowledge regarding the time sequences that lead to the event of interest. For example, in order to obtain ES_3 , pipe c should break within the initial 100 days whereas pipe b should work at least for 800 days after the failure of the first one.

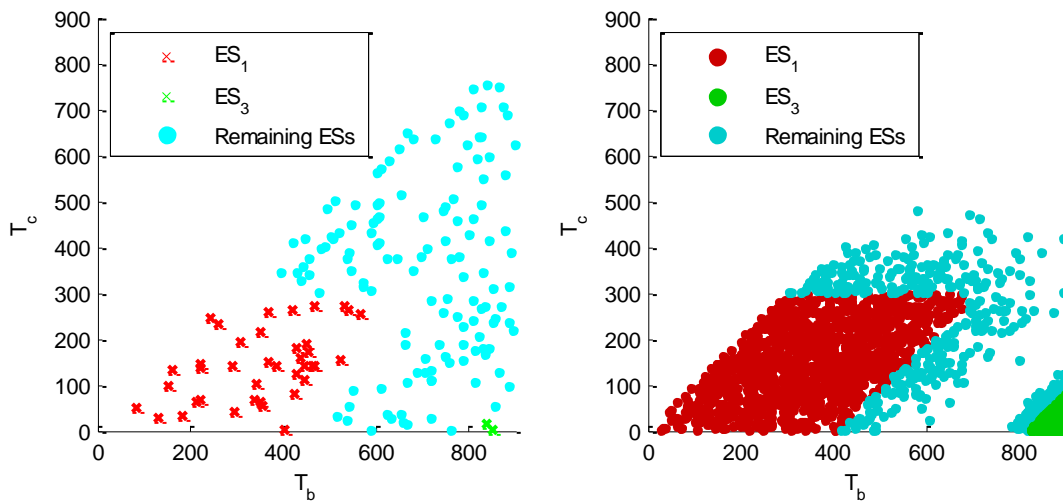


Figure 8 Preliminary guided exploration of S_5 (left) and deep exploration of ES_1 and ES_3 in the same scenario (right).

4.1.3 Discussion

Discovering and understanding the possible outcomes of accident progression, leaving out as little as possible of unexpected, adds significant value to a risk assessment. The proposed adaptive simulation framework guides the exploration of the accident scenarios towards those that show the highest variability in their outcomes, thus increasing the possibility of discovering a priori unexpected situations. The method allows including analyst prior knowledge regarding the accident scenarios and his/her preference towards specific outcomes to look for, making the method very flexible. In addition, new driving functions can be design for attaining specific objectives during the exploration, e.g., guiding the simulations towards the most risky scenarios.

Some weak points still remain in the proposed framework: i) it is assumed that the analyst is already aware of the accident scenarios that the system can undergo, which is not always the case in large systems involving a large number of components; nevertheless, some methods have been developed to

automatically generate possible risk scenarios (Li et al., 2011); ii) the proposed framework, in its present formulation, is not designed for parallel computing: however, by selecting and simulating batch of time sequences, it is possible to benefit from parallel computational resources.

4.2 CRITICAL REGIONS IDENTIFICATION

4.2.1 Method

With reference to Section 3, let us assume that a mathematical model $Y = f(\mathbf{X})$ of the system behavior is available, whose input $\mathbf{X} \in D_{\mathbf{X}} \subset \mathbb{R}^M$, represents a given system operational configuration and whose output $Y \in D_{\mathbf{Y}} \subset \mathbb{R}$ reflects the condition/state of the system. We define the conditions where $Y \geq Y_{thres}$ as “critical” and the corresponding configurations of inputs as the CR, i.e., $CR = \{\mathbf{x} \in D_{\mathbf{X}} \subset \mathbb{R}^M: y = f(\mathbf{x}) \geq Y_{thres}\}$. From a mathematical perspective, we are looking for the solution of the inverse problem $\mathbf{x} = f^{-1}(y)$, with $y \geq Y_{thres}$; however, this is not viable in the majority of the engineering systems where $f(\mathbf{x})$ is a function embedded in numerical codes, which is: *i)* complex, *ii)* black-box *iii)* not invertible.

A solution is, then, to resort to a DOE for exploring the I/O relation by means of numerical simulations and, then, retrieve information concerning the CRs through post-processing (Levy & Steinberg, 2010; Santner et al., 2003). However, this approach is hard to pursue when models have the characteristics mentioned in Section 2.

In what follows, a self-adaptive algorithm for exploring the numerical model and retrieving information regarding the CRs is presented. Eventual probabilistic distributions associated to \mathbf{X} are not considered, since the focus is, instead, on its range of values (i.e., on its domain), in order to explore all possible configurations during the CRs research. Hence, hereafter, without loss of generality, we assume that all inputs are standardized, e.g., $\mathbf{X} \in D_{\mathbf{X}} = [0,1]^M$ (Rosenblatt, 1952); likewise, a standardization can be applied to the output Y . This helps in designing a general, problem-independent algorithm and in removing effects related to the different orders of magnitudes possibly existing among inputs.

The driving idea of the proposed framework is to iteratively: *i)* run a (possibly small) number of model simulations, *ii)* retrieve knowledge from the available simulations and *iii)* guide the selection of new configurations towards the regions of interest (Turati, Pedroni, & Zio, 2016b). The framework is characterized by four principal steps (see Figure 9). In short, the first step aims at identifying the inputs that most affect the output of the model in order to limit the exploration only to the corresponding subspace (dimensionality reduction) resorting to PCE-based sensitivity analysis (Sudret, 2008). The second step aims at training a computationally cheap-to-run meta-model that accurately reproduces the response of the real model on the reduced space, with a particular attention to its capacity of discriminating between the CRs and normal conditions, e.g., a Kriging meta-model (Kleijnen, 2009). The third step resorts to the meta-model for deeply exploring the reduced state space by means of MCMC, with the objective of visiting and, consequently, discovering those configurations of inputs leading to critical outputs (Andrieu & Thoms,

2008). Finally, the last step employs clustering (e.g., k-means (Jain, 2010)) and graphical representation techniques (e.g., Parallel Coordinates Plot PCP (Inselberg, 2009)) for retrieving information and describing the CRs found.

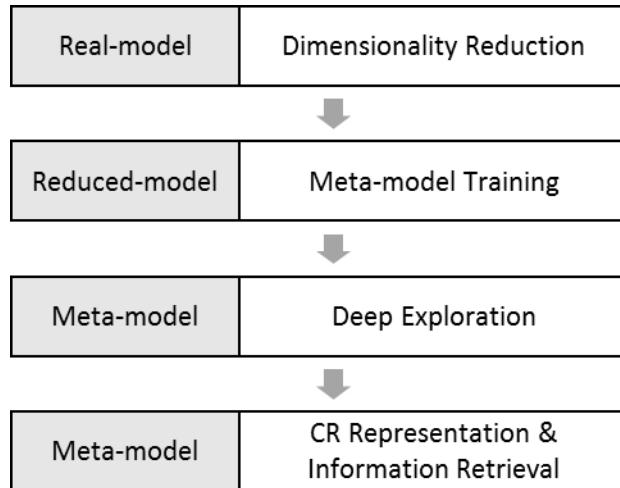


Figure 9 Flow diagram of the exploration framework.

4.2.1.1 Dimensionality Reduction

In general terms, dimensionality reduction includes a number of strategies for identifying a lower-dimensional subspace of variables where it is possible to build a reduced and simplified, yet representative and understandable, model of the system behavior (Fodor, 2002; H. Liu & Motoda, 2012). From the point of view of the exploration, reducing the dimensionality of the state space to explore allows the definition of a more effective DOE. Two main strategies have been proposed in the literature: *i*) feature selection, which aims at selecting a subset of the available variables and parameters input to the model (Guyon & Elisseeff, 2003), and *ii*) feature extraction, which aims at identifying a subset of “new” features created by means of transformations of the initial ones (Guyon & Elisseeff, 2006). Nevertheless, dimensionality reduction methods usually rely on a large set of input/output data examples that are not usually available, when the system model is computationally expensive.

In alternative, sensitivity analysis methods can be employed to achieve the same final objective as feature selection, by ranking the inputs according to their influence on the output of the model (Borgonovo & Plischke, 2016; Saltelli, 2008; Sudret, 2008). In particular, to this aim, global order sensitivity indices are more appropriate than local sensitivity indices, because they provide a measure of how the inputs globally affect the output of the model, i.e., with respect to different configurations of the inputs. In this paper, we resort to the total order sensitivity index S_T (Homma & Saltelli, 1996; Sobol, 2001) that is a variance-based

global sensitivity measure, assessing the expected fraction of the total variance of the output Y that is due to the variation of a specific input i and to its interactions with the others:

$$S_{Ti} = \frac{E_{\mathbf{X}_{\sim i}}[V_{\mathbf{X}_i}(Y|\mathbf{X}_{\sim i})]}{V(Y)}, \quad (5)$$

where \mathbf{X}_i represents the i -th component of the input vector \mathbf{X} , $\mathbf{X}_{\sim i}$ represents the rest of the components of the vector \mathbf{X} and $S_T \in [0,1]$. A large value of S_{Ti} indicates that the i -th input heavily affects Y and, thus, should be kept in what is hereafter called “reduced-model”; on the contrary, a very low value of S_{Ti} indicates that the i -th input does not affect Y and, thus, it can be discarded or set to a constant value. Usually, a threshold $S_{thres} = 1/M$ is adopted to discriminate the important inputs (Saltelli, 2008).

Although S_T usually requires a large number of MC or Quasi Monte Carlo (QMC) simulations to be accurately computed (Saltelli, 2008), PCE has been shown to achieve the same accuracy with a much lower number of simulations (Sudret, 2008) (see Appendix B, for details). For this reason, PCE is here employed to identify those inputs that must be kept in the reduced-model. All the analyses involving both the PCE approximation and the corresponding computation of the sensitivity indices, are conducted using the UQLab Toolbox for Matlab (Marelli & Sudret, 2014).

4.2.1.2 Meta-modeling

The main objective of a meta-model is to reproduce the behavior of the real (typically long-running) system model with a less expensive computational model. The meta-model is trained by resorting to a typically limited number of I/O observations from the real reduced model; on this basis, it should be capable of predicting the output values associated to input configurations that have not been explored yet. Since the real model is assumed to be deterministic (i.e., simulations of the same input configuration lead to the same output), it is desirable that the meta-model predicts as well the exact output value in correspondence of the training configurations (i.e., those known with absolute certainty). In this respect, among the numerous methods available in the literature (Jin, Chen, & Simpson, 2001; Shan & Wang, 2010), we resort to Kriging (Kleijnen, 2009; Matheron, 1963), i.e., Gaussian process modeling (see Appendix C for details). Kriging is capable of modeling local behaviors of the response function and of diversifying the levels of accuracy of the same model within different regions.

For example, in this case, the meta-model should be accurate in discriminating whether a configuration belongs to CR or not. For this reason, the meta-model should be more refined in the proximity of the CRs, whereas it can be rough in the rest of the space. To achieve this goal, sequential adaptive training strategies have been recently developed (Bect, Ginsbourger, Li, Picheny, & Vazquez, 2012; Echard, Gayton, & Lemaire,

2011; Picheny, Ginsbourger, Roustant, Haftka, & Kim, 2010). Instead of resorting to a static DOE to select the input/output configurations, new configurations are iteratively added to the training set to minimize a proper cost function. The Adaptive Kriging-Monte Carlo Simulation (AK-MCS) (Echard et al., 2011) is here employed to this aim.

In the AK-MCS, an initial Kriging model is trained with a small set of I/O observations, e.g., sampled according to LHS scheme; then, the algorithm proceeds iteratively according to the following steps: *i)* randomly sample a large set of input configurations $\mathcal{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_{MCS})})$, e.g., by means of LHS; *ii)* evaluate the associated responses using the Kriging meta-model $\hat{\mathcal{Y}} = (\hat{\mathcal{Y}}_1, \dots, \hat{\mathcal{Y}}_{N_{MCS}})$; *iii)* check if a convergence criterion has been reached: if so, the meta-model is sufficiently accurate; otherwise, *iv)* select, according to a predefined learning function/criterion, the best candidate subset $\mathcal{X}^* \subset \mathcal{X}$ to add to the current DOE and evaluate the corresponding real model output \mathcal{Y}^* ; *v)* retrain a new Kriging meta-model by adding the $\{\mathcal{X}^*, \mathcal{Y}^*\}$ to the training set and go back to step *i)*.

As learning function (step *iv* above), we consider the so-called U -function, which is based on the concept of misclassification (Echard et al., 2011):

$$U(\mathbf{x}) = \frac{|Y_{thres} - \mu_{\hat{\mathcal{Y}}}(\mathbf{x})|}{\sigma_{\hat{\mathcal{Y}}}(\mathbf{x})}. \quad (6)$$

In practice, $U(\mathbf{x})$ represents the distance in terms of standard deviations of the meta-model prediction from the limit state Y_{thres} . The smaller the value, the closer the prediction is to the limit state and, thus, the higher the interest in adding the corresponding I/O observation to the training set, because it reduces the prediction uncertainty regarding configurations “close” to the limit surface (in a probabilistic sense). Theoretically, the best DOE is obtained by adding at each iteration only one best candidate configuration. However, this increases the computational cost related to the training of the meta-model, which can be significant when a large number of I/O configurations are used and/or when many parameters have to be estimated due to the high dimensionality.

To overcome this problem, a larger number of I/O configurations can be added at the same time to the training set. Due to the correlation function, prediction points that are close share similar prediction values and misclassification probabilities; thus, it is likely that in the best candidate set, there are configurations having similar input values. However, evaluating the real model with respect to similar configurations increases the computational cost without adding the desired amount of knowledge to the meta-model. To this aim, clustering techniques are here employed to select, among the best candidate set, the most representative configurations before evaluating the corresponding real model output (Schöbi, Sudret, &

Marelli, 2016). An alternative method for optimally adding multiple observations to the training set has been recently proposed in (Chevalier et al., 2014).

As a stopping criterion (step *iii* above), we resort to the leave-one-out estimate of the correction factor $\hat{\alpha}_{corr\ LOO}$ (Dubourg, Sudret, & Deheeger, 2013):

$$\hat{\alpha}_{corr\ LOO} = \frac{1}{N_{Krig}} \sum_{n=1}^{N_{Krig}} \frac{\mathbb{1}_{f(\mathbf{x}^{(n)}) \geq Y_{thres}}(\mathbf{x}^{(n)})}{P(\hat{Y}_{DOE \setminus \mathbf{x}^{(n)}}(\mathbf{x}^{(n)}) \geq Y_{thres})}, \quad (7)$$

where $\hat{Y}_{DOE \setminus \mathbf{x}^{(n)}}(\mathbf{x}^{(n)})$ is the prediction of the output associated to the inputs $\mathbf{x}^{(n)}$, obtained with a Kriging model having as training set all the I/O observations except $(\mathbf{x}^{(n)}, y_n)$. This verifies that the probabilistic discriminating function (i.e., the prediction) converges towards the real discriminating function (i.e., the real limit surface). In practice, a value of $\hat{\alpha}_{corr\ LOO}$ close to 1 indicates a satisfactory approximation of the real model, whereas very small or very large values indicate an inaccurate approximation. It must be noticed that, since the estimation is based on a Leave-One-Out (LOO) cross-validation, a minimum number of initial I/O observations, (e.g., 30 (Dubourg et al., 2013)), has to be provided to guarantee accurate estimates. On the other side, a maximum number of iterations can be set, in order to limit the number of calls to the real model.

For building the meta-model, we resort to the UQLab Toolbox for Matlab (Marelli & Sudret, 2014), whereas the sequential training algorithm has been developed by the authors.

4.2.1.3 Deep Exploration

During the Deep EXploration (DEX) phase, the aim is to exploit the meta-model, to thoroughly explore the system space, in particular, to discover possible unexpected CRs. An algorithm based on the MCMC M-H algorithm has been designed. Although we refer the reader to the corresponding paper (Turati et al., 2016b), we list here the main ideas. The iterative algorithm, at each step, firstly identifies the number of CRs already discovered using clustering techniques; then, several Markov Chains are distributed among the CRs in order to guarantee that each CR has been explored with the same meticulousness. In practice, the CRs having a low density of simulation runs within them are more likely to be underexplored than those having a higher density: thus, more Markov Chains will be assigned to the underexplored regions. For each one of the configuration visited by the Markov Chains, the corresponding meta-model is evaluated and if it leads to a critical output, it is added to the CRs. The algorithm continues until the number of CRs identified remains equal for a given number of iterations, i.e., until no more new CRs are identified, or alternatively

until a certain density of simulations is reached for all the CRs. In any case a maximum number of simulations can be set for controlling the maximum computational effort.

4.2.1.4 Critical Regions Representation & Information Retrieval

The outcome of the deep exploration is typically a large dataset containing a large set of points belonging to several CRs. However, when the state space dimensionality is higher than 3-4 dimensions, high-dimensional data visualization techniques are necessary to retrieve useful insights. The interested reader is referred to (S. Liu, Maljovec, Wang, Bremer, & Pascucci, 2015) for an extended review of the state of the art. In what follows, we make use of two of the most known techniques: ScatterPLOT Matrix (SPLOM) (Hartigan, 1975) and the PCP (Inselberg, 2009), which help in retrieving complementary information about the CRs, such as their shapes and the corresponding input values in a unique, “readable”, graphical representation.

4.2.1.5 Exploration Assessment

Assuming that the real limit function representing the configurations in the CRs is available, the objective of the assessment phase is to measure how satisfactorily the exploration method has identified the configurations leading to critical conditions. Only for illustrative purposes, Figure 10 left shows the output of an accurate exploration of a two-dimensional space, where the real CR (shaded) is sufficiently covered by the configurations selected by the explorative method (circles); on the contrary, Figure 10 right shows an incomplete exploration where a fraction of a CR is identified, but not entirely covered, and another CR is not even explored.

Quantitative metrics are here introduced to assess the quality of the exploration: in particular, the population of critical configurations visited by the proposed methodology \mathcal{X}_{exp}^{CR} (circles) is compared to a uniformly distributed population of samples belonging to the real CRs \mathcal{X}_{real}^{CR} (crosses), according to a distance-based criterion.

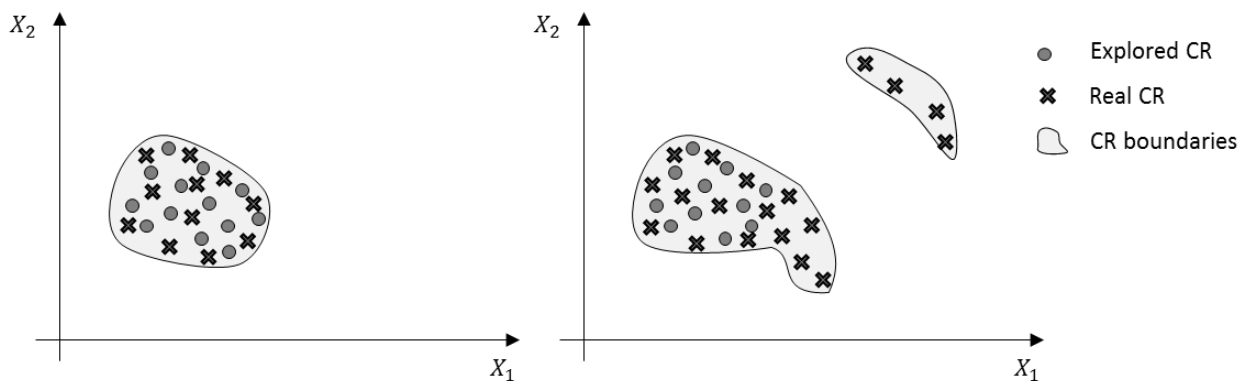


Figure 10 Representation of an accurate CR exploration (left) and of an incomplete CR exploration (right).

A one-vs-all version of the Local Outlier Factor (LOF) is employed to this aim, where each configuration in the real CRs is compared to the whole population of critical configurations obtained by the exploration method. For the sake of completeness, LOF is a density-based outlier detection method capable of measuring how isolated is a sample from the rest of a given population of interest (Breunig, Kriegel, Ng, & Sander, 2000). In our case, the more isolated a real CR configuration is from the explored ones, the higher the probability that it belongs to an unexplored CR.

The definition of the LOF relies on the concept of reachability distance between points \mathbf{x} and \mathbf{o} :

$$d_{reach}(\mathbf{x}, \mathbf{o}) = \max(d_{kNN}(\mathbf{o}), d(\mathbf{x}, \mathbf{o})), \quad (8)$$

where $d(\cdot; \cdot)$ is a generic distance and $d_{kNN}(\mathbf{o})$ is the distance of the k Near Neighbor (kNN) of \mathbf{o} . In this paper, the Euclidean distance is employed; however, the Manhattan or even lower order L^p distances can be preferable in high dimensionality (Aggarwal, Hinneburg, & Keim, 2001). Then, the local reachability distance, which measures how close is the configuration \mathbf{x} to its $kNNs$, can be defined as:

$$lrd_k(\mathbf{x}) = \frac{k}{\sum_{\mathbf{o} \in kNN(\mathbf{x})} d_{reach}(\mathbf{x}, \mathbf{o})}. \quad (9)$$

In this light, the LOF of a configuration \mathbf{x} is defined as:

$$LOF(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{o} \in kNN(\mathbf{x})} \frac{lrd_k(\mathbf{o})}{lrd_k(\mathbf{x})}, \quad (10)$$

where the parameter k has to be set by the analyst (and it is not related to the number of clusters K identified in Section 4.2.1.4).

In general, a value of $LOF(\mathbf{x}) \approx 1$ indicates that the configuration \mathbf{x} is well represented by the rest of the configurations, whereas a value of $LOF(\mathbf{x}) \gg 1$ indicates that the configuration \mathbf{x} is isolated. In order to have a reference value for detecting a critical configuration as unexplored, the LOF is evaluated for all critical configurations $\mathbf{x} \in \mathcal{X}_{exp}^{CR}$ (namely, LOF_{exp}). Likewise, LOF_{real} represents the random variables corresponding to the one-vs-all evaluations of the configurations $\mathbf{x} \in \mathcal{X}_{real}^{CR}$. A configuration $\mathbf{x} \in \mathcal{X}_{real}^{CR}$ is considered “unexplored”, if $LOF(\mathbf{x}) > \overline{LOF}_{exp}$, where:

$$\overline{LOF}_{exp} = \max_{\mathbf{x} \in \mathcal{X}_{exp}^{CR}} LOF(\mathbf{x}) \quad (11)$$

is the LOF corresponding to the most isolated configuration explored.

The following distance-based statistics have been considered to synthetize the overall performance of the exploration method:

1. Expected LOF:

$$\mu_{LOF}^{real} = E[LOF_{real}] \quad (12)$$

A value of $\mu_{LOF}^{real} \gg 1$ indicates that some CRs are probably unexplored.

2. Unexplored Critical Region (UCR):

$$UCR = \frac{\#(LOF_{real} > \overline{LOF}_{exp})}{\#\mathcal{X}_{real}^{CR}} \quad (13)$$

which is the ratio between the number of real critical configurations identified as unexplored and the cardinality of \mathcal{X}_{real}^{CR} . In practice, it represents the “fraction” of CRs that have not been explored by the method.

3. Unexplored Extreme Critical Region (UECR):

$$UECR_{\gamma\%} = UCR_{\gamma\%} | \mathcal{X}_{real}^{ECR} = \frac{\#(LOF_{real} > \overline{LOF}_{exp} | \mathcal{X}_{real}^{ECR})}{\#\mathcal{X}_{real}^{ECR}} \quad (14)$$

where $\mathcal{X}_{real}^{ECR} \subset \mathcal{X}_{real}^{CR}$ is the subset of the CRs leading to the most “extreme” outputs. In particular, $\gamma \in [0,100]\%$ is the quantile used to characterize the extreme outputs: letting $\gamma = 0.9$, then a critical configuration is considered “extreme” if its output is larger than the output of 90% of the population. This metric allows the analyst to understand whether the method has discovered the CRs leading to the most critical outputs.

4. Conditional Expected LOF:

$$\mu_{LOF|UCR} = E \left[\frac{LOF_{real}}{\overline{LOF}_{exp}} \mid LOF_{real} > \overline{LOF}_{exp} \right] \quad (15)$$

that indicates how much isolated are on average the unexplored critical configurations with respect to the most isolated critical configuration explored. In practice, values of $\mu_{LOF|UCR} \gg 1$ indicate the presence of critical configurations that are very isolated from the explored CRs and, thus, warn the analyst on the presence of CRs disconnected from those already identified.

4.2.2 Power Distribution Network

A power distribution network is analyzed in order to discover its associated CRs (Mena, Hennebel, Li, Ruiz, & Zio, 2014). The network, represented in Figure 11, is composed of 10 feeders transporting energy from a unique Main Source (MS) to 8 demanding nodes (consumers) characterized by different daily load profiles.

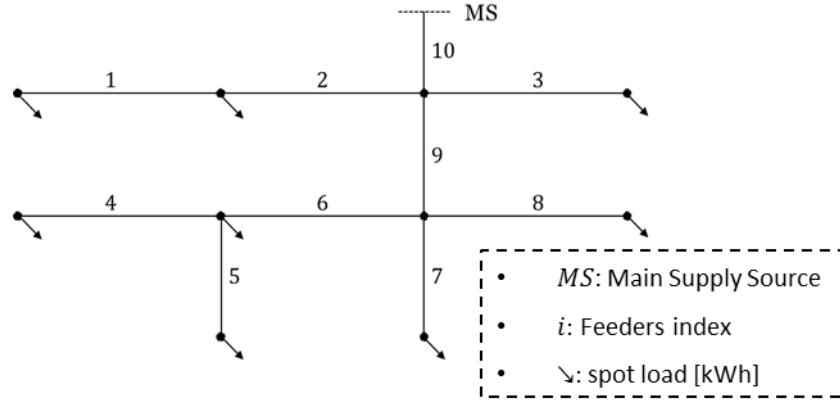


Figure 11 Power network configuration.

The load profiles L_j assume different shapes according to the corresponding type of consumers associated. These include residential consumers and offices, whose per unit (p.u.) daily spot load profiles are reported in Figure 12. In detail, the daily load L_j of a demanding node is given by:

$$L_j(t) = r_j R(t) + o_j O(t) \quad (16)$$

where $\mathbf{R}(t)$ and $\mathbf{O}(t)$ are the p.u. daily loads, whereas \mathbf{r}_j and \mathbf{o}_j are the corresponding average loads for the residential consumer and office, respectively (Jardini, Tahan, Gouvea, Ahn, & Figueiredo, 2000). The values of the average loads used in this paper are reported in Table VI. Uncertainty and seasonality effects on the average loads can be easily embedded into the model. Nevertheless, since the focus of the study is on the exploration of the daily profiles to verify the impact of feeder failures, they are not taken into account in the analysis.

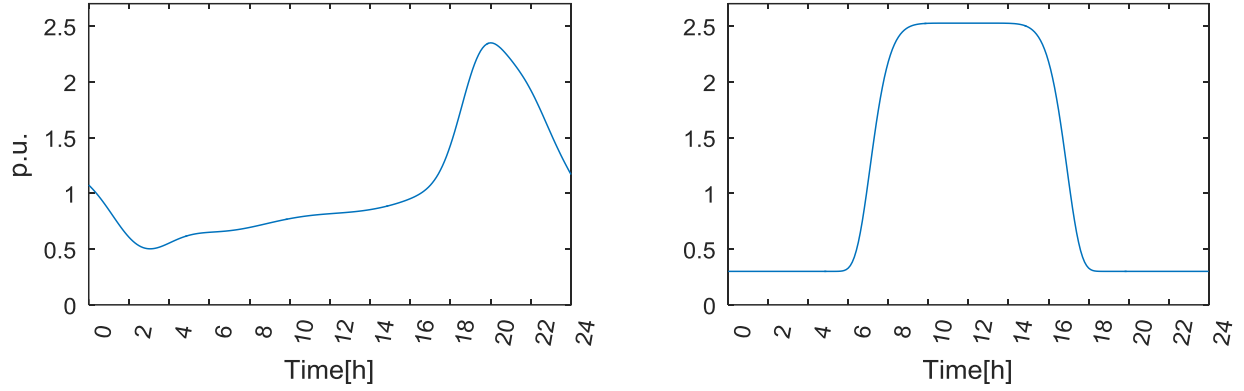


Figure 12 Power load profiles for a residential consumer (left) and for a commercial office (right).

Table VI Average load values for the 10 nodes of the network in kW.

Node	1	2	3	4	5	6	7	8	9	10
R	0	0	0	1	1	5	5	5	0	0
O	5	5	100	0	0	0	0	0	0	0

We assume that each feeder can independently fail only once within the 24 hour, at a random time $T_i \in [0,24)$ and with associated magnitude of the failure F_i . When the i -th feeder fails, no power can flow through it for a time proportional to the magnitude of the failure: for example, $F_i = 0.5$ means that the feeder is out of service for half an hour. In this view, $\mathbf{X} = [T_1, \dots, T_{10}, F_1, \dots, F_{10}]$ is the M-dimensional vector of the inputs to the model and represents a given failure configuration.

The electrical Energy Not Served (ENS) to the consumers is considered as output of the model and it is defined in this case as:

$$ENS(\mathbf{X}) = \int_0^{24} \sum_{i=1}^{10} \mathbf{1}_{NSS(t)}(i) \cdot L_i(t) dt, \quad (17)$$

where $NSS(t)$ indicates the Not Supplied Set at time t , i.e., the set of nodes that are not served at time t and $\mathbf{1}$ is the indicator function, which takes value 1 if $i \in NSS(t)$ and 0 otherwise. Moreover, ENS is used to discriminate the critical conditions, i.e., a value of $ENS(\mathbf{X}) \geq ENS_{thres}$ implies that the failure configuration \mathbf{X} is critical; otherwise, \mathbf{X} is considered as “normal”. The value of ENS_{thres} is set equal to 500 kWh, in order to focus the attention on critical events.

4.2.2.1 Dimensionality Reduction

For the dimensionality reduction step, we resort to PCE, where the maximum degree of the polynomials is fixed to 5 in order to reduce the computational cost and focus the attention on the main trend of the model. The coefficients of the PCE are estimated by Least Angle Regression on the basis of a DOE of 500 samples obtained with a QMC Sobol' sequence (Sobol, Asotsky, Kreinin, & Kucherenko, 2011). Figure 13 shows that there is a huge difference between the total order indices S_T of the inputs: those associated to feeders 3 and 10 (i.e., T_3, T_{10}, F_3, F_{10}) take values larger than 0.2, whereas the others take values lower than 0.05. This is in accordance with the fact that feeders 3 and 10 are the only two that can affect the energy supplied to the most demanding consumer (i.e., user 3). In this light, the dimensionality of the reduced-model is set to 4 with $\mathbf{X}^* = (T_3, T_{10}, F_3, F_{10})$, and the rest of the inputs are set to randomly fixed values, since they are expected to have no effect on the output.

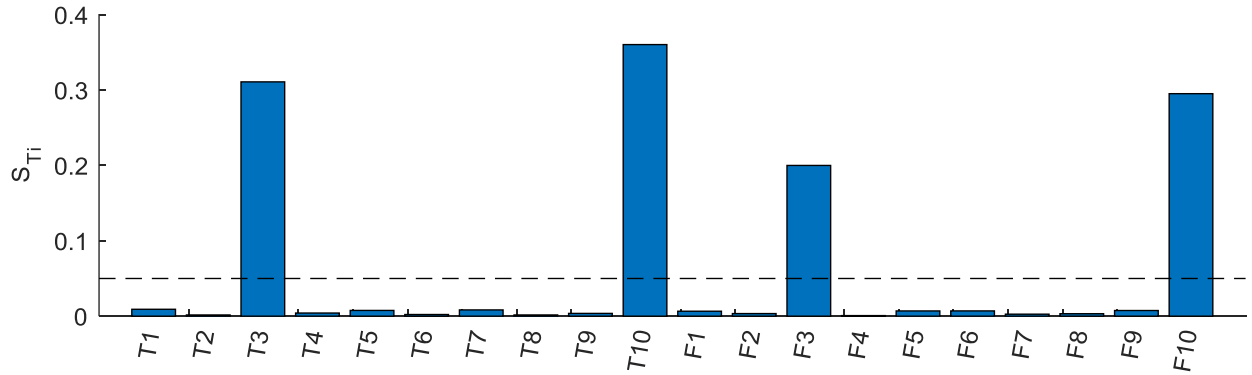


Figure 13 Sobol' total order indices for the 20 inputs.

4.2.2.2 Meta-model

For training the meta-model we resort to an ordinary kriging, i.e., the trend is assumed to be unknown but constant, which allows the Gaussian process to completely adapt to the training data. An ellipsoidal anisotropic correlation function is used to take into account possible different behaviors of the response function with respect to different inputs: in particular, we resort to the 3/2 Matérn one (Abramowitz & Stegun, 1964; Rasmussen & Williams, 2006):

$$h(x, x'; \boldsymbol{\theta}) = \sqrt{\sum_{m \in M'} \left(\frac{x_m - x'_m}{\theta_m} \right)^2} \quad (18)$$

$$R\left(h, v = \frac{3}{2}\right) = (1 + h\sqrt{3}) \cdot e^{-h\sqrt{3}}$$

where v is the shape parameter and $\boldsymbol{\theta}$ the scale one.

Given the dimensionality of the reduced-model, 100 configurations sampled with a Sobol' QMC and the corresponding ENS are used for initializing the meta-model. Then, through the iterative AK-MCS introduced in Section 4.2.1.2, 10000 configurations are sampled by means of LHS and a maximum of 50 candidate configurations are evaluated and added to the DOE $\{\mathbf{x}_{krig}, \mathbf{y}_{krig}\}$ at each step. Only configurations having a value of the U-function lower than 4 are eligible as candidates. Actually, $U(\mathbf{x}) > 4$ indicates that the corresponding configuration is, in a probabilistic view, very distant from the critical threshold. A maximum number of 1000 I/O observations for training the meta-model is set in order to limit the maximum computational effort. Figure 14 shows the projection on the two-dimensional subspace $[T_3, T_{10}]$ of the configurations used to train the meta-model: on the left panel, we report the initial 100 samples used for the initialization, whereas on the right, those added iteratively by the AK-MCS are shown. It is worth noticing how the adaptive DOE distributes the observations differently in the different portions of the input domain (i.e., a significantly higher density in the CRs).

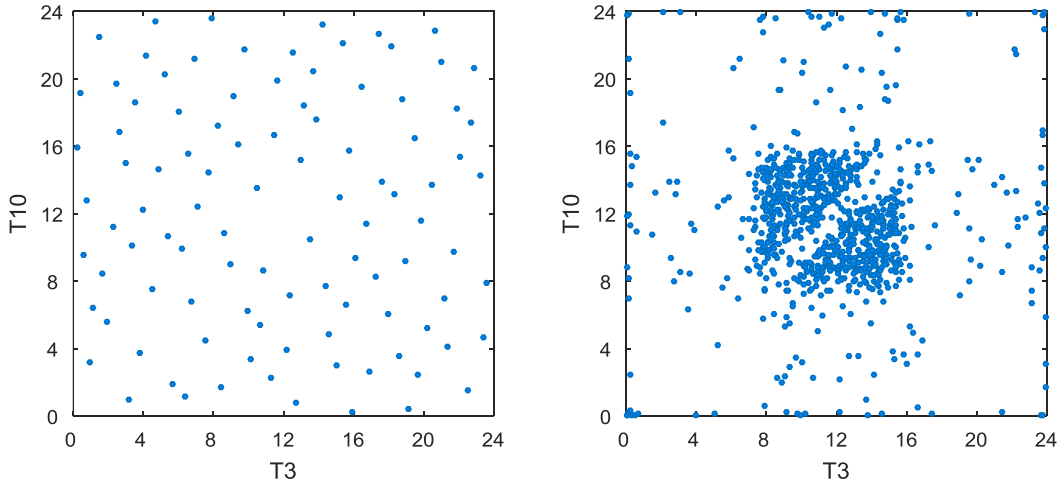


Figure 14 Projection of the DOE used for training the meta-model. The Figure on the left shows the initial 100 Sobol' QMC samples, whereas on the right those added by the AK-MCS are shown.

4.2.2.3 Deep Exploration

From the Kriging DOE, 169 configurations are identified as critical. In order to deeply explore the CRs, 5 iterations of the method proposed in Section 4.2.1.3 are run with 5 Markov chains and a maximum number of samples equal to 5000. Figure 15 shows the projections on the two-dimensional subspace $[T_3, T_{10}]$ of the configurations belonging to the CRs. The left panel reports the configurations available from the meta-model DOE, whereas that on the right contains those obtained as a result of the deep exploration (~3000 configurations). It is worth noticing that the deep exploration allows better highlighting the boundaries of the CRs and, thus, to better retrieve their shapes and characteristics. This is even more apparent in high-

dimensional spaces. Only one projection of the CRs configurations is here reported for brevity; nevertheless, a detailed analysis is given in the following sections.

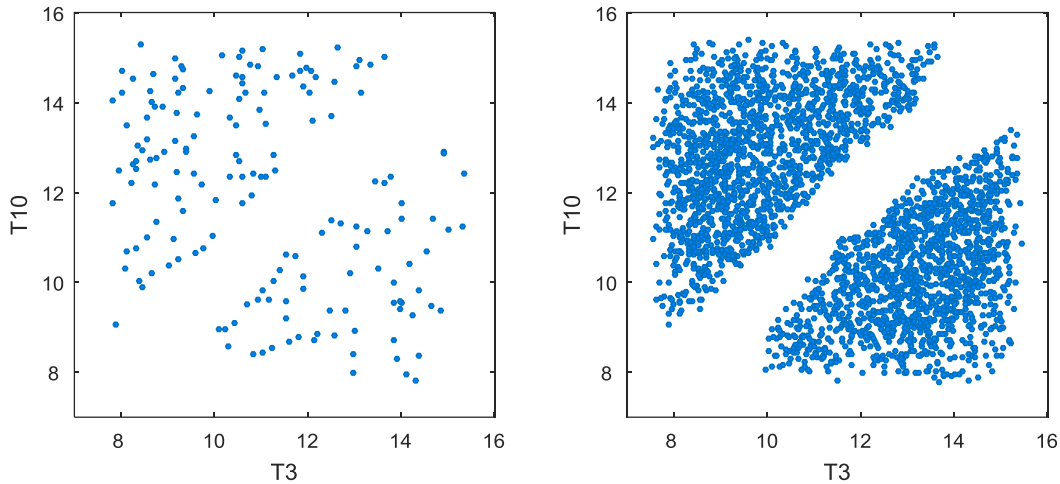


Figure 15 Two-dimensional projections of the observations belonging to the CRs: those available from the DOE of the meta-model (left) and those obtained with the deep exploration step (right).

4.2.2.4 Representation & Information Retrieval

A sequence of k-means clusterings with different cluster cardinality (from $K=1$ to 10) is applied to the critical configurations for identifying the representative number of separate CRs. Several cluster validity indices (e.g., Hubert statistic, Dunn, Silhouette, Davies and Bouldin, Calinski and Harabasz indices, etc.) have been computed to this aim; however, since this analysis goes beyond the present scope, the reader is referred to (Arbelaitz, Gurrutxaga, Muguerza, Pérez, & Perona, 2013; Charrad, Ghazzali, Boiteau, & Niknafs, 2014) for details on the definition and interpretation of the indices used. Two clusters have been identified and the corresponding PCP is reported in Figure 16. For the sake of clarity, the envelopes of the parallel coordinates representing the two clusters (i.e., the ranges of values characterizing the clusters) are shown in Figure 17. By observing these ranges, it is also possible to have an idea of the dimension of the CRs. In this case, for example, they occupy respectively around the (30%, 30%, 20%, 20%) of the entire range of the four important inputs T_3 , T_{10} , F_3 and F_{10} , which corresponds to $\sim 0.36\%$ of the entire input domain. The CRs are characterized by failures occurring during the central hours of the day (between 8-15) and with a failure magnitude above the 0.8, i.e., the feeders are out of order for at least 48min each. In addition, it is worth noticing that the two clusters show different behaviors on the two axes corresponding to the failure times, i.e., T_3 and T_{10} .

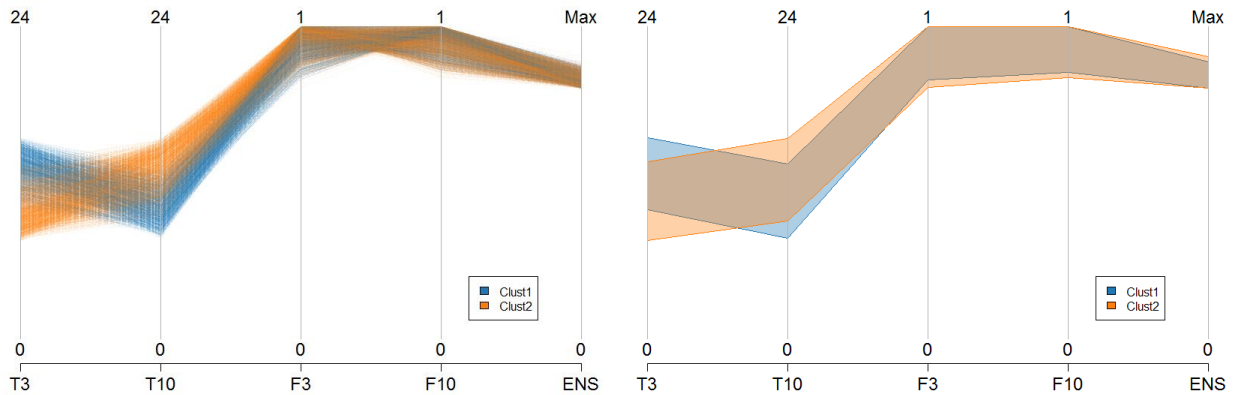


Figure 16 Parallel Coordinates Plot of the two CRs identified. **Figure 17** Envelopes of the PCP representing the inputs ranges.

For this reason, the corresponding SPOLM is given in Figure 18, where the “envelopes” identified on the PCP are represented in the panels above the diagonal by means of shadowed rectangles. It can be observed that the two clusters are recognizable and well separated on the subspace defined by $[T_3, T_{10}]$: cluster 1 is characterized by an initial failure of feeder 10 followed by a failure of feeder 3 with a delay of at least one hour, whereas cluster 2 is characterized by the inverse sequence, still with a delay of at least one hour between failures. Indeed, if both failures happen at the same time, the ENS associated to node 3 is the same as if only one of the two failures had happened, because both feeders are put under repair at the same time and, thus, the total time of energy not supplied to user 3 is “just” one hour.

Concerning the subspace defined by $[F_3, F_{10}]$, it must be noticed that there is no difference between the two clusters. However, the triangular shape of the region shows that the sum of the two failure magnitudes must be at least equal to 1.80, i.e., the consumer at node 3 is not served for at least 1h:48m.

Finally, although the two-dimensional projections of the PCP envelopes overestimate the regions of the associated CRs, they provide a synthetic representation, which can be useful as first approximation of the CRs.

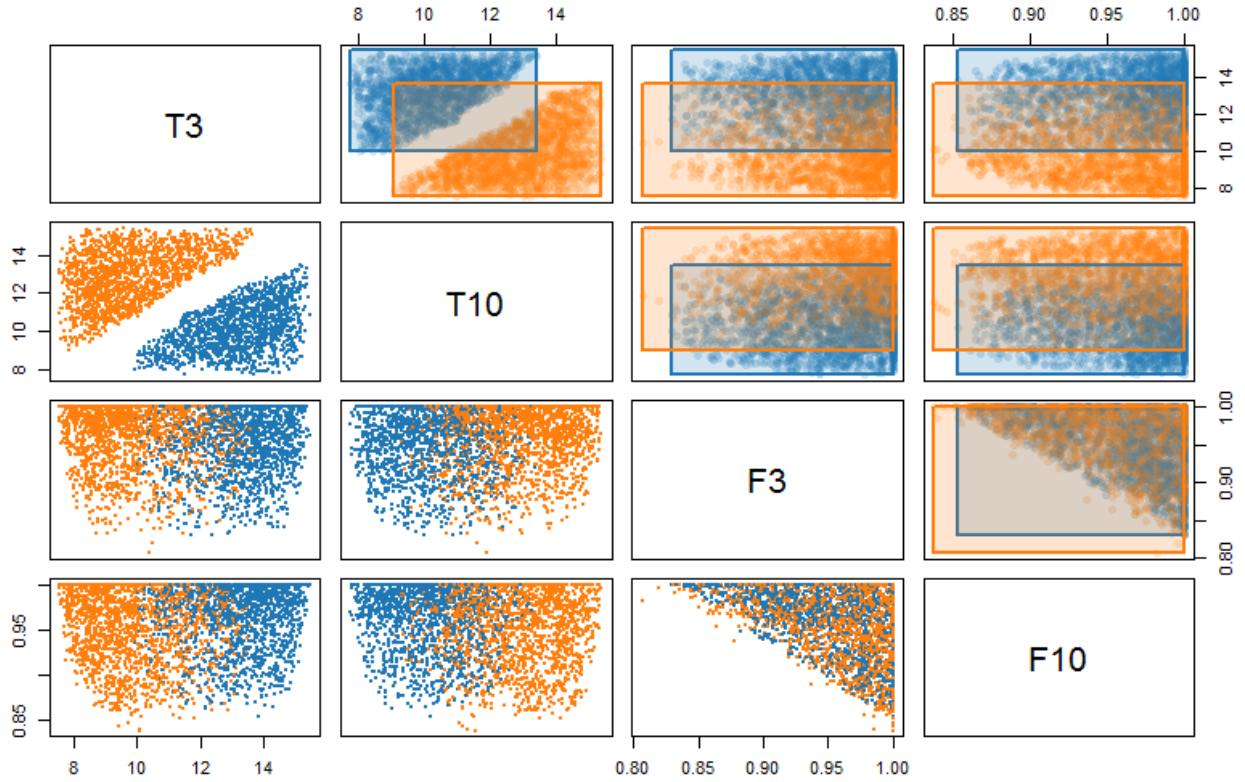


Figure 18 SPLOM of the two CRs discovered by the exploration algorithm. Above the diagonal, the projections of the PCP envelopes are depicted by means of shadowed rectangles.

4.2.2.5 Performance Assessment

In order to have a representative picture of the real CRs, a large number of configurations involving all 20 inputs of the model have been sampled by means of LHS and the corresponding output has been evaluated. Moreover, the outputs of the reduced-model involving the projections of the 20 inputs on the 4-dimensional space defined by $[T_3, T_{10}, F_3, F_{10}]$ have been evaluated as representative of the ideal “target”, meta-model representation. The number of calls to the expensive model and/or to the cheap one (i.e., the meta-model) is given in Table VII for each exploration strategy.

Table VII Number of calls made to the computationally cheap and/or expensive model for the different exploration strategies.

COPUTATIONAL COST	META-MODEL	REDUCED-MODEL	REAL-MODEL
CHEAP	~200000	0	0
EXPENSIVE	1500	100000	100000

Among the large number of configurations sampled, those leading to critical values of ENS are selected and the corresponding LOF evaluated to verify to what extent the CRs discovered by the meta-model are

similar to those found by the reduced and real-models (see Section 4.2.1.5). The values of the associated statistics are given in Table VIII. The CRs of the meta-model are used as the reference set, thus, only the corresponding expected value of the LOF can be evaluated. By looking at the results obtained for the reduced-model, it must be observed that all the statistics assume low values: the average value of LOF is very close to that of the meta-model; the percentage of CRs that remains unexplored is only 3%, and the associated conditional value is still very low (i.e., 1.08), which means that the unexplored CRs are very close to the boundaries of the CRs identified by the meta-model. In this light, it can be stated that the meta-model exploration has accurately explored and discovered the CRs associated to the reduced-model.

On the other side, with respect to the real model, the average LOF takes a large value compared to the meta-model, suggesting that a part of the CRs remains unexplored. This is confirmed by the percentage of unexplored CRs. However, it must be noticed that the percentage of unexplored extreme CRs is very low, i.e., the meta-model exploration has been able to identify the configurations leading to the most critical outputs. Finally, the conditional expected value $\mu_{LOF|UCR}$ takes a value that is not very large, suggesting that the unexplored portion of CRs is likely to be close to the boundaries.

Table VIII Local Outlier Factor (LOF)-based statistics for the different exploration strategies.

METRIC	META-MODEL	REDUCED-MODEL	REAL-MODEL
μ_{LOF}	1.02	1.03	2.66
UCR	-	3%	72%
$UECR_{90\%}$	-	0%	7%
$\mu_{LOF UCR}$	-	1.08	2.20

In order to visualize the results, we resort to a SPLOM where the CRs identified by the meta-model exploration are depicted by light circles and the configuration belonging to the CRs associated to the real model are depicted by crosses and squares according to their values of LOF. In particular, in accordance with Section 4.2.1.5, those configurations having $LOF \leq \overline{LOF}_{exp}$ (see Eq. (11)) are defined as identified CRs (crosses), whereas those having $\overline{LOF}_{exp} < LOF$ are defined as undiscovered CRs (squares). It must be noticed that there is not a significant difference between the Meta-Model (MM)-based and the real model-based exploration in the subspace characterized by the failure times $[T_3, T_{10}]$. On the contrary, there is a significant difference in the failure magnitude subspace $[F_3, F_{10}]$: according to the real model, it is enough that the sum of the magnitudes is larger than ~ 1.60 . This means that the real model can reach a critical condition even if the consumer at node 3 is not served for at least 1h:36m. Indeed, the rest of the ENS

needed to reach the critical threshold can come from the failures of the feeders discarded during the dimensionality reduction step. Finally, by looking at the last column of Figure 19, it can be seen that the largest values of ENS, i.e., the most critical ones, are correctly discovered by our methodology (crosses).

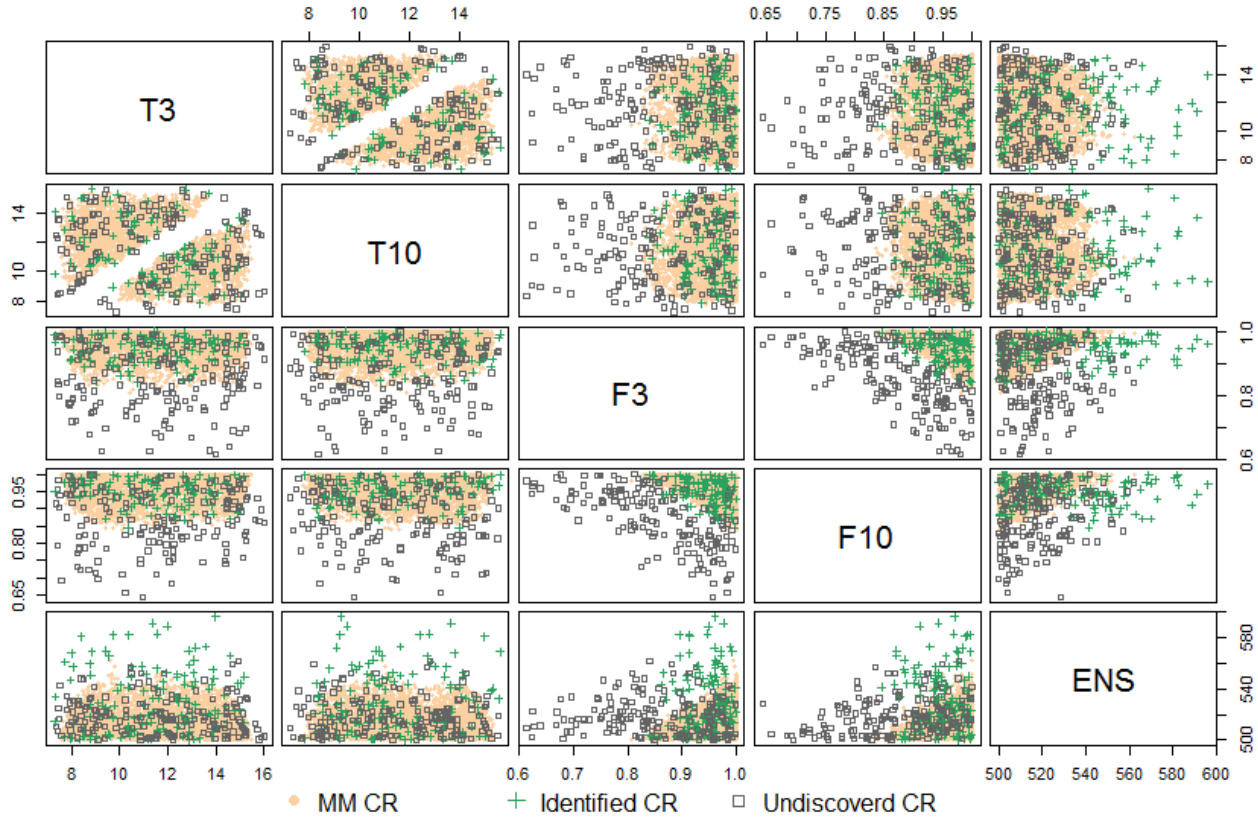


Figure 19 SPLOM of the CRs discovered by the Meta-Model (MM) exploration (light circles). The CRs of the real model are depicted with different symbols whether identified (cross) or not (square).

A sort of sensitivity analysis to the model parameters has also been conducted to verify the performance of the proposed methodology when the impacts of the discarded inputs is very low, i.e., when the reduced-model is likely to represent the real model. To this aim, all the loads except that of node 3 have been reduced of a factor 10 (the corresponding values are reported in Table IX). In order to assure the presence of a CR despite the loading reduction, the threshold ENS_{thres} has been set equal to 475 kWh, i.e., 5% lower than the initial one. All the analyses have been run with the same settings and with the same number of calls to the model as in the initial case.

Table IX Average load values for the 10 nodes of the network in kW.

NODE	1	2	3	4	5	6	7	8	9	10
R	0	0	0	0.1	0.1	0.5	0.5	0.5	0	0
O	0.5	0.5	100	0	0	0	0	0	0	0

Table X reports the result of the statistics associated to the LOF for the reduced and the real model-based exploration. The average value of the LOF is for all types of exploration very close to 1, indicating that it is likely that all CRs have been discovered. This is confirmed by the percentage of unexplored CRs, which is null for both models. The value of $\mu_{LOF|UCR}$ is not reported, since no configuration has been identified as unexplored.

Table X Local Outlier Factor (LOF)-based statistics for the different exploration strategies.

METRIC	META-MODEL	REDUCED-MODEL	REAL-MODEL
μ_{LOF}	1,02	1,01	1,07
UCR	-	0	0
$UECR_{90\%}$	-	0	0

Figure 20 shows that all critical configurations discovered by means of the real model-based exploration (dark crosses) lay inside or at the boundaries of the CRs discovered by the proposed methodology (light circles). These results demonstrate how the proposed methodology is capable of identifying the CRs resorting to a limited number of calls to the real model: in this case, two orders of magnitude lower than the exploration based on the real model.

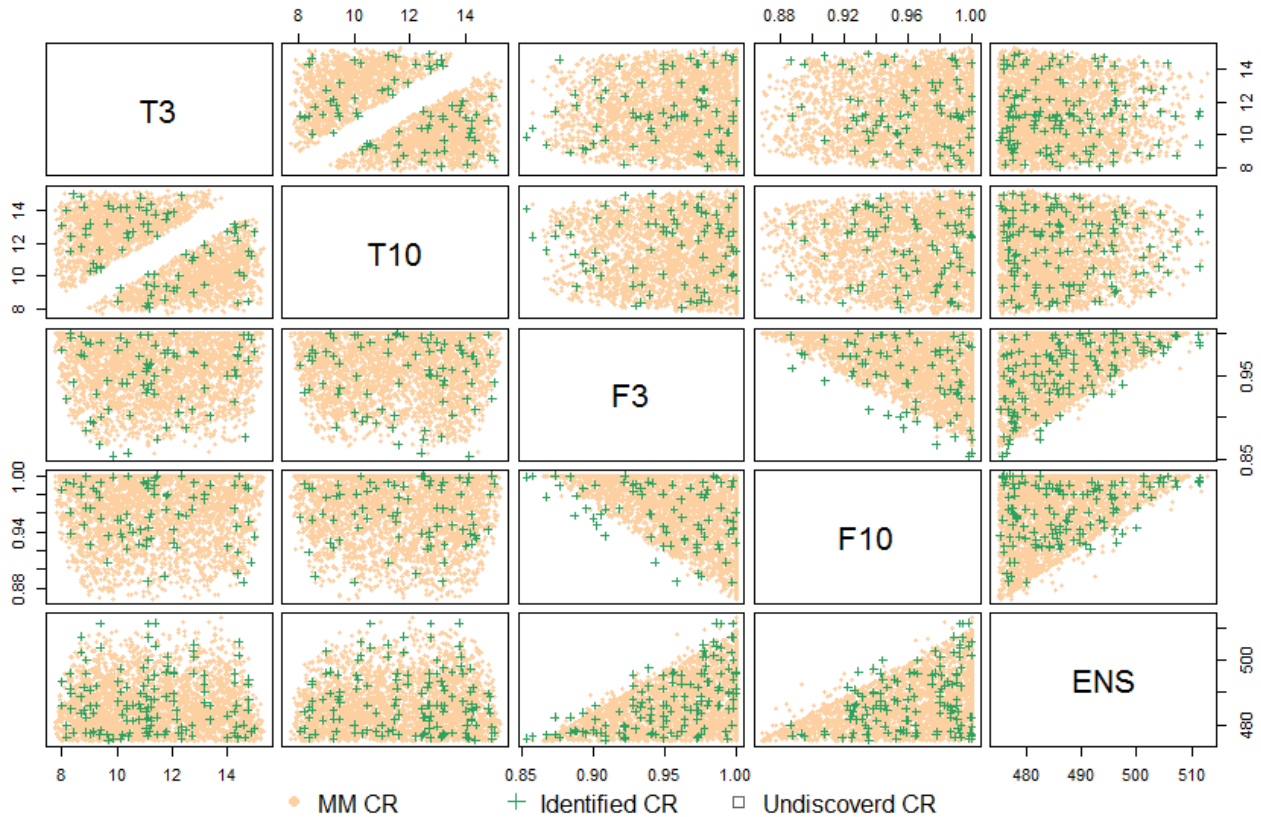


Figure 20 SPLOM of the CRs discovered by the Meta-Model (MM) exploration (light circles). The CRs of the real model are depicted with different symbols, whether identified (cross) or not (square).

4.2.3 Discussion

In Section 4.2 a new strategy has been proposed to identify and characterize CRs by simulations of models that are: i) computationally expensive, ii) high-dimensional, iii) complex.

The main advantage of the proposed method is the capability of exploring and retrieving information with a limited number of simulations. Furthermore, the method is general and modular, i.e., it can be applied to a variety of problems and cases. For example, if the numerical model is not high-dimensional (or computationally expensive), the dimensionality reduction step (or the meta-model one) can be avoided.

Finally, since the proposed method relies on the capability of the meta-model of accurately reproducing the behavior of the real model, the performance of the method is in a way conditioned by that of the Kriging. In particular, Kriging performance tends to decrease with the dimensionality of the important input space, i.e., the dimensionality of the reduced-model input space.

5 Conclusions

In this chapter, the possibility of gaining knowledge for system risk assessment by scenario simulations has been discussed and investigated. The trivial idea is to explore how the system behaves by running simulations and retrieving a posteriori the information of interest, specifically with respect to those unexpected or unusual critical configurations forming the so-called CRs. Such exploration becomes obviously challenging, when the simulation model is: *i)* high-dimensional; *ii)* complex; *iii)* black box and *iv)* computationally expensive. Specific methods are, then, needed to obtain the information of interest with a limited number of calls to the (computationally expensive) model. Two main strategies have been considered in the literature to this aim. One resorts to parallel computing to reduce the time required to achieve a satisfactory level of detail during the exploration. The other one resorts to iterative adaptive strategies, which exploit the knowledge available from the results of the simulations already run, to select the best configuration for a new “informative” simulation (which should, in principle, add more information on the states of the system that are of interest for the analysis).

Two methods have been presented in the chapter. One explores the uncertainty associated to different possible accident scenarios in order to increase the knowledge about the impact that time has on the evolution of scenarios of interest. The method identifies those scenarios characterized by a large variability in their output and, consequently, concentrates the simulation runs on them. At the same time, the method can embed the prior knowledge of the analyst. This allows focusing the attention and the majority of the computational efforts on the exploration of a limited number of accident scenarios.

The other method aims at identifying and characterizing the configurations of inputs and parameters leading a system to abnormal conditions, i.e., those of the CRs. The proposed framework makes use of: *i)* dimensionality reduction techniques, to limit the dimensionality of the input space; *ii)* meta-modeling to reproduce the real model and reduce the computational cost for a model run; *iii)* an adaptive exploration algorithm to identify and thoroughly probe the critical regions; *iv)* clustering and high-dimensional data visualization techniques to retrieve and visualize the knowledge enclosed in the simulations run. The framework is modular and flexible, making it easy to adapt to different types of applications.

Finally, it must be emphasized that the knowledge that can be retrieved from the simulations is conditioned on the knowledge available in the model: the more detailed and accurate the model, the more challenging the exploration, but also the more complete and informative the information that can be retrieved.

6 References

- Abramowitz, M., & Stegun, I. A. (1964). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables* (Vol. 55): Courier Corporation.
- AEMO. (2016). *PRELIMINARY REPORT –BLACK SYSTEM EVENT IN SOUTH AUSTRALIA ON 28 SEPTEMBER 2016*. Retrieved from <http://www.aemo.com.au/Media-Centre/-/media/BE174B1732CB4B3ABB74BD507664B270.ashx>
- Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). *On the surprising behavior of distance metrics in high dimensional space*: Springer.
- Aldemir, T. (2013). A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants. *Annals of Nuclear Energy*, 52, 113-124. doi:DOI 10.1016/j.anucene.2012.08.001
- Alfonsi, A., Rabiti, C., Mandelli, D., Cogliati, J. J., Wang, C., Maljovec, D. P., . . . Smith, C. L. (2016). *RAVEN Theory Manual*. Retrieved from
- Andrieu, C., & Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18(4), 343-373. doi:DOI 10.1007/s11222-008-9110-y
- Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1), 243-256.
- Authority, E. B. (2016). *2016 EU-Wide Stress Test - Methodological Notes*. Retrieved from <http://www.eba.europa.eu/-/eba-launches-2016-eu-wide-stress-test-exercise>.
- Aven, T. (2012a). Foundational Issues in Risk Assessment and Risk Management. *Risk Analysis*, 32(10), 1647-1656. doi:10.1111/j.1539-6924.2012.01798.x
- Aven, T. (2012b). On the critique of Beck's view on risk and risk analysis. *Safety Science*, 50(4), 1043-1048. doi:<http://dx.doi.org/10.1016/j.ssci.2011.12.004>
- Aven, T. (2013). On the meaning of a black swan in a risk context. *Safety Science*, 57, 44-51. doi:<http://dx.doi.org/10.1016/j.ssci.2013.01.016>
- Aven, T. (2016a). Ignoring scenarios in risk assessments: Understanding the issue and improving current practice. *Reliability Engineering & System Safety*, 145, 215-220. doi:<http://dx.doi.org/10.1016/j.ress.2015.08.012>
- Aven, T. (2016b). Risk assessment and risk management: Review of recent advances on their foundation. *European Journal of Operational Research*, 253(1), 1-13. doi:<http://dx.doi.org/10.1016/j.ejor.2015.12.023>
- Aven, T., & Zio, E. (2014). Foundational Issues in Risk Assessment and Risk Management. *Risk Analysis*, 34(7), 1164-1172. doi:10.1111/risa.12132
- Baudin, M., Dutfoy, A., Iooss, B., & Popelin, A.-L. (2016). OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation. In R. Ghanem, D. Higdon, & H. Owhadi (Eds.), *Handbook of Uncertainty Quantification* (pp. 1-38). Cham: Springer International Publishing.
- Bect, J., Ginsbourger, D., Li, L., Picheny, V., & Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3), 773-793. doi:10.1007/s11222-011-9241-4
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9), 509-517. doi:10.1145/361002.361007
- Blatman, G., & Sudret, B. (2011). Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6), 2345-2367.
- Borgonovo, E., & Plischke, E. (2016). Sensitivity analysis: A review of recent advances. *European Journal of Operational Research*, 248(3), 869-887. doi:<http://dx.doi.org/10.1016/j.ejor.2015.06.032>
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). *LOF: identifying density-based local outliers*. Paper presented at the ACM sigmod record.

- Burnaev, E., Panin, I., & Sudret, B. (2016). Effective Design for Sobol Indices Estimation Based on Polynomial Chaos Expansions *Conformal and Probabilistic Prediction with Applications* (pp. 165-184): Springer International Publishing.
- Cepin, M., & Mavko, B. (2002). A dynamic fault tree. *Reliability Engineering & System Safety*, 75(1), 83-91. doi:Doi 10.1016/S0951-8320(01)00121-1
- Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *2014*, 61(6), 36. doi:10.18637/jss.v061.i06
- Cheng, B., & Titterton, D. M. (1994). Neural networks: A review from a statistical perspective. *Statistical science*, 2-30.
- Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., & Richet, Y. (2014). Fast Parallel Kriging-Based Stepwise Uncertainty Reduction With Application to the Identification of an Excursion Set. *Technometrics*, 56(4), 455-465. doi:10.1080/00401706.2013.860918
- Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *American Statistician*, 49(4), 327-335. doi:Doi 10.2307/2684568
- Clarke, S. M., Griebisch, J. H., & Simpson, T. W. (2004). Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. *Journal of Mechanical Design*, 127(6), 1077-1087. doi:10.1115/1.1897403
- Cojazzi, G. (1996). The DYLAM approach for the dynamic reliability analysis of systems. *Reliability Engineering & System Safety*, 52(3), 279-296.
- Commision, E. (2013). *Technical summary on the implementation of comprehensive risk and safety assessments of nuclear power plants in the European Union* (SWD287). Retrieved from [http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52012SC0287R\(01\)&from=EN](http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52012SC0287R(01)&from=EN).
- Cox, A. L. J. (2015). Special (virtual) Issue: Foundations of Risk Analysis.
- Di Maio, F., Bandini, A., Zio, E., Alfonsi, A., & Rabiti, C. (2016). An approach based on Support Vector Machines and a K-D Tree search algorithm for identification of the failure domain and safest operating conditions in nuclear systems. *Progress in Nuclear Energy*, 88, 297-309. doi:<http://dx.doi.org/10.1016/j.pnucene.2016.01.017>
- Di Maio, F., Baronchelli, S., & Zio, E. (2015a). A Computational Framework for Prime Implicants Identification in Noncoherent Dynamic Systems. *Risk Analysis*, 35(1), 142-156. doi:Doi 10.1111/Risa.12251
- Di Maio, F., Baronchelli, S., & Zio, E. (2015b). A Visual Interactive Method for Prime Implicants Identification. *IEEE Transactions on Reliability*, 64(2), 539-549. doi:10.1109/TR.2014.2371015
- Di Maio, F., Vagnoli, M., & Zio, E. (2015). Risk-Based Clustering for Near Misses Identification in Integrated Deterministic and Probabilistic Safety Analysis. *Science and Technology of Nuclear Installations*, 501, 693891.
- Dreossi, T., Dang, T., Donzé, A., Kapinski, J., Jin, X., & Deshmukh, J. V. (2015). Efficient Guiding Strategies for Testing of Temporal Properties of Hybrid Systems. In K. Havelund, G. Holzmann, & R. Joshi (Eds.), *NASA Formal Methods: 7th International Symposium, NFM 2015, Pasadena, CA, USA, April 27-29, 2015, Proceedings* (pp. 127-142). Cham: Springer International Publishing.
- Dubourg, V., Sudret, B., & Deheeger, F. (2013). Metamodel-based importance sampling for structural reliability analysis. *Probabilistic Engineering Mechanics*, 33, 47-57. doi:<http://dx.doi.org/10.1016/j.probengmech.2013.02.002>
- Echard, B., Gayton, N., & Lemaire, M. (2011). AK-MCS: An active learning reliability method combining Kriging and Monte Carlo Simulation. *Structural Safety*, 33(2), 145-154. doi:<http://dx.doi.org/10.1016/j.strusafe.2011.01.002>
- Eldred, M. S., Hough, P. D., Hu, K. T., Jakeman, J. D., Stephens, J. A., Swiler, L. P., . . . Wildey, T. M. (2014). *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter*

- Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual.*
Retrieved from
- Fainekos, G. E., Sankaranarayanan, S., Ueda, K., & Yazarel, H. (2012). *Verification of automotive control applications using S-TaLiRo*. Paper presented at the Proceedings of the American Control Conference.
- Fang, K.-T., Li, R., & Sudjianto, A. (2005). *Design and modeling for computer experiments*. Boca Raton: CRC Press.
- Flage, R., & Aven, T. (2015). Emerging risk – Conceptual definition and a relation to black swan type of events. *Reliability Engineering & System Safety*, 144, 61-67. doi:<http://dx.doi.org/10.1016/j.ress.2015.07.008>
- Fodor, I. (2002). A Survey of Dimension Reduction Techniques.
- Garrett, C., & Apostolakis, G. (1999). Context in the risk assessment of digital systems. *Risk Analysis*, 19(1), 23-32.
- Ghanem, R. G., & Spanos, P. D. (1991). *Stochastic finite elements: a spectral approach*. Berlin: Springer.
- Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., & Crombecq, K. (2010). A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of machine learning research*, 11(Jul), 2051-2055.
- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar), 1157-1182.
- Guyon, I., & Elisseeff, A. (2006). An Introduction to Feature Extraction. In I. Guyon, M. Nikravesh, S. Gunn, & L. A. Zadeh (Eds.), *Feature Extraction: Foundations and Applications* (pp. 1-25). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hakobyan, A., Aldemir, T., Denning, R., Dunagan, S., Kunsman, D., Rutt, B., & Catalyurek, U. (2008). Dynamic generation of accident progression event trees. *Nuclear Engineering and Design*, 238(12), 3457-3467. doi:DOI 10.1016/j.nucengdes.2008.08.005
- Hartigan, J. A. (1975). Printer graphics for clustering. *Journal of Statistical Computation and Simulation*, 4(3), 187-213. doi:10.1080/00949657508810123
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97-109. doi:10.1093/biomet/57.1.97
- Haykin, S., & Network, N. (2004). A comprehensive foundation. *Neural Networks*, 2(2004).
- Homma, T., & Saltelli, A. (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1), 1-17. doi:Doi 10.1016/0951-8320(96)00002-6
- Hsueh, K. S., & Mosleh, A. (1996). The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. *Reliability Engineering & System Safety*, 52(3), 297-314. doi:Doi 10.1016/0951-8320(95)00140-9
- Hu, Y., Groen, F., & Mosleh, A. (2004). *An entropy-based exploration strategy in dynamic PRA*. Paper presented at the Probabilistic Safety Assessment and Management.
- Ibáñez, L., Hortal, J., Queral, C., Gómez-Magán, J., Sánchez-Perea, M., Fernández, I., . . . Villalba-Jabonero, E. (2016). Application of the Integrated Safety Assessment methodology to safety margins. Dynamic Event Trees, Damage Domains and Risk Assessment. *Reliability Engineering & System Safety*, 147, 170-193. doi:<http://dx.doi.org/10.1016/j.ress.2015.05.016>
- Inselberg, A. (2009). *Parallel coordinates*: Springer.
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651-666. doi:<http://dx.doi.org/10.1016/j.patrec.2009.09.011>
- Jardini, J. A., Tahan, C., Gouvea, M., Ahn, S. U., & Figueiredo, F. (2000). Daily load profiles for residential, commercial and industrial low voltage consumers. *Power Delivery, IEEE Transactions on*, 15(1), 375-380.

- Jin, R., Chen, W., & Simpson, T. W. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1), 1-13. doi:10.1007/s00158-001-0160-4
- Kernstine, K. H. (2012). *Design space exploration of stochastic system-of-systems simulations using adaptive sequential experiments*. Georgia Institute of Technology.
- Kleijnen, J. P. C. (2009). Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3), 707-716. doi:<http://dx.doi.org/10.1016/j.ejor.2007.10.013>
- Kloos, M., & Peschke, J. (2006). MCDET: A probabilistic dynamics method combining Monte Carlo simulation with the discrete dynamic event tree approach. *Nuclear Science and Engineering*, 153(2), 137-156.
- Kuhnt, S., & Steinberg, D. M. (2010). Design and analysis of computer experiments. *AStA Advances in Statistical Analysis*, 94(4), 307-309. doi:10.1007/s10182-010-0143-0
- Labeau, P. E., Smidts, C., & Swaminathan, S. (2000). Dynamic reliability: towards an integrated platform for probabilistic risk assessment. *Reliability Engineering & System Safety*, 68(3), 219-254. doi:Doi 10.1016/S0951-8320(00)00017-X
- Le Matre, O. P., Reagan, M. T., Najm, H. N., Ghanem, R. G., & Knio, O. M. (2002). A stochastic projection method for fluid flow. II. Random process. *Journal of Computational Physics*, 181(1), 9-44. doi:10.1006/jcph.2002.7104
- Levy, S., & Steinberg, D. M. (2010). Computer experiments: a review. *AStA Advances in Statistical Analysis*, 94(4), 311-324. doi:10.1007/s10182-010-0147-9
- Li, J. H., Kang, R., Mosleh, A., & Pan, X. (2011). Simulation-based automatic generation of risk scenarios. *Journal of Systems Engineering and Electronics*, 22(3), 437-444. doi:DOI 10.3969/j.issn.1004-4132.2011.03.011
- Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining* (Vol. 454): Springer Science & Business Media.
- Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., & Pascucci, V. (2015, 15/11/2015). *Visualizing high-dimensional data: Advances in the past decade*. Paper presented at the Proc. Eurographics Conf. Visualization.
- Maio, F. D., Secchi, P., Vantini, S., & Zio, E. (2011). Fuzzy C-Means Clustering of Signal Functional Principal Components for Post-Processing Dynamic Scenarios of a Nuclear Power Plant Digital Instrumentation and Control System. *IEEE Transactions on Reliability*, 60(2), 415-425. doi:10.1109/TR.2011.2134230
- Mandelli, D., Smith, C., Rabiti, C., Alfonsi, A., Youngblood, R., Pascucci, V., . . . Zamalieva, D. (2013). *Dynamic PRA: An overview of new algorithms to generate, analyze and visualize data*. Paper presented at the Transactions of the American Nuclear Society.
- Mandelli, D., Yilmaz, A., Aldemir, T., Metzroth, K., & Denning, R. (2013). Scenario clustering and dynamic probabilistic risk assessment. *Reliability Engineering & System Safety*, 115, 146-160. doi:<http://dx.doi.org/10.1016/j.ress.2013.02.013>
- Marelli, S., & Sudret, B. (2014). *UQLab: a framework for uncertainty quantification in MATLAB*. Paper presented at the International Conference on Vulnerability, Risk Analysis and Management (ICVRAM2014), Liverpool (United Kingdom).
- Matheron, G. (1963). Principles of geostatistics. *Economic geology*, 58(8), 1246-1266.
- McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2), 239-245. doi:10.1080/00401706.1979.10489755
- Mena, R., Hennebel, M., Li, Y.-F., Ruiz, C., & Zio, E. (2014). A risk-based simulation and multi-objective optimization framework for the integration of distributed renewable generation and storage. *Renewable and Sustainable Energy Reviews*, 37, 778-793.

- Montero-Mayorga, J., Queral, C., & Gonzalez-Cadelo, J. (2014). Effects of delayed RCP trip during SBLOCA in PWR. *Annals of Nuclear Energy*, 63, 107-125. doi:10.1016/j.anucene.2013.06.030
- Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2016). *Response surface methodology: process and product optimization using designed experiments*: John Wiley & Sons.
- Nghiem, T., Sankaranarayanan, S., Fainekos, G., Ivanci, F., Gupta, A., & Pappas, G. J. (2010). *Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems*. Paper presented at the Proceedings of the 13th ACM international conference on Hybrid systems: computation and control, Stockholm, Sweden.
- Patelli, E., Broggi, M., Angelis, M. d., & Beer, M. (2014). *OpenCossan: An efficient open tool for dealing with epistemic and aleatory uncertainties*. Paper presented at the Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management.
- Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., & Kim, N.-H. (2010). Adaptive Designs of Experiments for Accurate Approximation of a Target Region. *Journal of Mechanical Design*, 132(7), 071008-071008. doi:10.1115/1.4001873
- Queral, C., Mena-Rosell, L., Jimenez, G., Sanchez-Perea, M., Gomez-Magan, J., & Hortal, J. (2016). Verification of SAMGs in SBO sequences with Seal LOCA. Multiple damage domains. *Annals of Nuclear Energy*, 98, 90-111. doi:<http://dx.doi.org/10.1016/j.anucene.2016.07.021>
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*: the MIT Press.
- RELAP5-3D. (2005). RELAP5-3D. Idaho Falls, Idaho: Idaho National Laboratory.
- Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). New York: Springer.
- Roberts, G. O., & Rosenthal, J. S. (2009). Examples of Adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2), 349-367. doi:DOI 10.1198/jcgs.2009.06134
- Rosenblatt, M. (1952). Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics*, 23(3), 470-472.
- Rutt, B., Catalyurek, U., Hakobyan, A., Metzroth, K., Aldemir, T., Denning, R., . . . Kunsman, D. (2006). Distributed dynamic event tree generation for reliability and risk assessment. *Challenges of Large Applications in Distributed Environments, Proceedings*, 61-70.
- Saltelli, A. (2008). *Global sensitivity analysis : the primer*. Chichester, England ; Hoboken, NJ: John Wiley.
- Santner, T. J., Williams, B. J., & Notz, W. (2003). *The Design and analysis of computer experiments*. New York: Springer.
- Schöbi, R., Sudret, B., & Marelli, S. (2016). Rare Event Estimation Using Polynomial-Chaos Kriging. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, D4016002.
- Shan, S., & Wang, G. G. (2010). Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2), 219-241. doi:10.1007/s00158-009-0420-2
- Simpson, T. W., Poplinski, J. D., Koch, N. P., & Allen, J. K. (2001). Metamodels for Computer-based Engineering Design: Survey and recommendations. *Engineering with Computers*, 17(2), 129-150. doi:10.1007/pl00007198
- Siu, N. (1994). Risk Assessment for Dynamic-Systems - an Overview. *Reliability Engineering & System Safety*, 43(1), 43-73. doi:Doi 10.1016/0951-8320(94)90095-7
- Smidts, C., & Devooght, J. (1992). Probabilistic Reactor Dynamics .2. A Monte-Carlo Study of a Fast-Reactor Transient. *Nuclear Science and Engineering*, 111(3), 241-256.
- Sobol. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3), 271-280. doi:Doi 10.1016/S0378-4754(00)00270-6
- Sobol, Asotsky, D., Kreinin, A., & Kucherenko, S. (2011). Construction and Comparison of High-Dimensional Sobol'Generators. *Wilmott*, 2011(56), 64-79.

- Soize, C., & Ghanem, R. (2004). Physical Systems with Random Uncertainties: Chaos Representations with Arbitrary Probability Measure. *SIAM Journal on Scientific Computing*, 26(2), 395-410. doi:10.1137/s1064827503424505
- Sorge, M. (2004). *Stress-testing Financial Systems: An Overview of Current Methodologies*. Retrieved from SSRN: <http://ssrn.com/abstract=759585> or <http://dx.doi.org/10.2139/ssrn.759585>
- Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7), 964-979.
- Taleb, N. N. (2007). *The black swan: The impact of the highly improbable*: Random house.
- Turati, P., Pedroni, N., & Zio, E. (2015). *An entropy-driven method for exploring extreme and unexpected accident scenario in the risk assessment of dynamic engineered systems*. Paper presented at the Proceedings of the 25th ESREL, Safety and Reliability of Complex Engineered Systems, Zurich, Swiss.
- Turati, P., Pedroni, N., & Zio, E. (2016a). An Adaptive Simulation Framework for the Exploration of Extreme and Unexpected Events in Dynamic Engineered Systems. *Risk Anal.* doi:10.1111/risa.12593
- Turati, P., Pedroni, N., & Zio, E. (2016b). *Simulation-Based Exploration Of High-Dimensional System Models For Critical Regions Identification*. Reliability Engineering & System Safety (Submitted).
- Wang, G. G., & Shan, S. (2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4), 370-380.
- Zio, E. (2014). Integrated deterministic and probabilistic safety assessment: Concepts, challenges, research directions. *Nuclear Engineering and Design*, 280, 413-419. doi:DOI 10.1016/j.nucengdes.2014.09.004
- Zio, E. (2016a). Challenges in the vulnerability and risk analysis of critical infrastructures. *Reliability Engineering & System Safety*, 152, 137-150. doi:10.1016/j.res.2016.02.009
- Zio, E. (2016b). Some challenges and opportunities in reliability engineering. *IEEE Transactions on Reliability*, (submitted).

Appendix

A. METROPOLIS-HASTINGS

The Metropolis-Hastings (M-H) is a well-known Markov Chain Monte Carlo (MCMC) method for sampling from unconventional probability distributions. The general idea of a MCMC method is to generate a Markov Chain having the *target* distribution p as its stationary distribution (Robert & Casella, 2004).

For generating the Markov Chain, the M-H algorithm iteratively samples a candidate \mathbf{T}^* from a *proposal distribution* q , and *accept-reject* the proposed sample according to an acceptance criterion (Hastings, 1970).

For the *proposal* step, easy to sample distributions are usually considered. For example, in Section 4 we resort to a Multivariate Gaussian distribution $q(\mathbf{T}^* | \mathbf{T}_n) \sim N(\mathbf{T}_n, \Sigma)$, having as mean value the last accepted sample \mathbf{T}_n and as covariance matrix Σ , whose coefficient can be estimated using a set of samples available from the *target* distribution, or can be set a priori by the analyst. Once sampled, the candidate \mathbf{T}^* can be accepted (i.e., $\mathbf{T}_{n+1} = \mathbf{T}^*$) or rejected (i.e., $\mathbf{T}_{n+1} = \mathbf{T}_n$) with a probability $\alpha(\mathbf{T}_n, \mathbf{T}^*) = \min(r(\mathbf{T}_n, \mathbf{T}^*), 1)$, where r is defined as follows:

$$r(\mathbf{T}_n, \mathbf{T}^*) = \begin{cases} \frac{p(\mathbf{T}^*) \cdot q(\mathbf{T}_n | \mathbf{T}^*)}{p(\mathbf{T}_n) \cdot q(\mathbf{T}^* | \mathbf{T}_n)}, & p(\mathbf{T}_n) \cdot q(\mathbf{T}^* | \mathbf{T}_n) > 0 \\ 1, & otherwise \end{cases}, \quad (19)$$

p being the *target* distribution from which we want to sample. If the *proposal* distribution is symmetric, i.e., $q(\mathbf{T}_n | \mathbf{T}^*) = q(\mathbf{T}^* | \mathbf{T}_n)$, then, Eq. (19) can be rewritten as:

$$r(\mathbf{T}_n, \mathbf{T}^*) = \begin{cases} \frac{p(\mathbf{T}^*)}{p(\mathbf{T}_n)}, & p(\mathbf{T}_n) > 0 \\ 1, & otherwise. \end{cases} \quad (20)$$

Finally, if the target distribution is uniform on the support Ω_I of the event of interest, then, the probability $\alpha(\mathbf{T}_n, \mathbf{T}^*)$ can be written as:

$$\alpha(\mathbf{T}_n, \mathbf{T}^*) = \begin{cases} 1, & \mathbf{T}^* \in \Omega_I \\ 0, & otherwise. \end{cases} \quad (21)$$

In order to reach with a small number of samples the stationary distribution, a critical indicator is the Acceptance Ratio (AR) between the proposed candidate and the accepted ones: if AR is too high (AR>0.9), it is likely that the proposed candidate is very close to the previous one, meaning that the Markov Chain is too slow in spanning the space of interest. On the contrary, if AR is small (AR<0.2), the proposal distribution is sampling candidates that are too distant from the accepted ones and thus in regions where the target

distribution is very low or even outside the target domain Ω_I , meaning that distribution is approximated with several repetitions of the same samples.

B. POLYNOMIAL CHAOS EXPANSION – BASED SENSITIVITY ANALYSIS

Given a function $Y = f(\mathbf{X})$, where \mathbf{X} represents a vector of random inputs and Y is the associated output. It is possible to decompose the function by means of the Polynomial Chaos Expansion (PCE) representation (Ghanem & Spanos, 1991), that is:

$$Y = f(X_1, \dots, X_M) = \sum_{\alpha \in \mathbb{N}^M} y_\alpha \psi_\alpha(X_1, \dots, X_M), \quad (22)$$

where y_α is the coefficient associated to the multivariate Hilbertian basis $\psi_\alpha(\cdot)$, orthonormal with respect to the multivariate distribution characterizing the inputs (usually the uniform or the normal distribution are considered). In order to be valid, the Hilbertian space should be chosen such that it contains the response function Y (Soize & Ghanem, 2004). If the input multivariate distribution is uniform, then $\psi_\alpha(\cdot)$ is a multivariate Legendre polynomial, where the multi-index $\alpha = (\alpha_1, \dots, \alpha_M)$ indicates the order of the polynomials associated to each component of the vector \mathbf{X} . For example, if $\alpha = (3, 1, 0, 2)$, then the associated Legendre polynomial is characterized by a third order polynomial for X_1 , a first order polynomial for X_2 , a zero order polynomial for X_3 and a second order polynomial for X_4 . The polynomial chaos expansion, in order to keep reasonable the numerical cost, can be truncated to a maximum polynomial order p , providing an approximation of the real response function:

$$Y = f(X_1, \dots, X_M) \approx \sum_{\alpha \in A^{M,p}} y_\alpha \psi_\alpha(X_1, \dots, X_M), \quad (23)$$

where $A^{M,p} \subset \mathbb{N}^M$ is the multi-index subset corresponding to polynomials having maximum order equal to p , i.e., $A^{M,p} = \{\alpha \in \mathbb{N}^M \text{ s. t. } |\alpha| < p\}$ with corresponding cardinality $\#A^{M,p} = \binom{M+p}{p}$.

The great advantage of the PCE is that, once the approximation (23) is computed, then the total order sensitivity indices can be trivially approximated as:

$$S_{Ti} \approx \tilde{S}_{Ti} = \frac{\sum_{\mathbf{u} \in U^i} y_{\mathbf{u}}^2}{\sum_{\alpha \in A^{M,p}} y_{\alpha}^2}, \quad (24)$$

where $U^i = \{\mathbf{u} \in A^{M,p} \text{ s. t. } u_i \neq 0\}$ is the subset of all the multi-indices corresponding to multivariate Legendre polynomials with non-zero degree associated to the i -th component, i.e., the subset of multi-indices representing polynomials that include the i -th component (Sudret, 2008). The approximated total order sensitivity indices \tilde{S}_{Ti} converges to the real one with the degree of the polynomial truncation p . In practice,

the computational cost required for estimating S_T depends only on the computational cost needed to approximate the output function with the PCE.

The estimation of the PCE coefficients can be conducted both via projection and regression. Even though the projection technique is more rigorous, it requires to know explicitly the definition of the function f (Le Matre, Reagan, Najm, Ghanem, & Knio, 2002), which is typically not the case when dealing with black box functions or complex numerical codes. For this reason, we resort to a regression method, in particular, to the Least Angle Regression (LARS) coupled with an adaptive sparse PCE representation (Blatman & Sudret, 2011), which is devised to automatically detect the significant PCE coefficients limiting at the same time the computational cost for the PC approximation. The sparse representation of the coefficient matrix, indeed, allows keeping into the memory only those coefficients having a non-negligible value, which is typically the case in many real applications. In order to train the regression model, a number N_{PCE} of input configurations is usually sampled according to Latin Hypercube Sampling (LHS) or other Quasi Monte Carlo (QMC) techniques (McKay, Beckman, & Conover, 1979; Sobol et al., 2011). Consequently, the corresponding real model outputs are evaluated and used to fit the regression model. Recently, an optimal DOE for the estimation of the PC coefficients has been proposed to further reduce the number of calls to the possibly long-running model (Burnaev, Panin, & Sudret, 2016).

Finally, it must be pointed out that PCE is a meta-modeling technique capable of well representing the global behavior of the response function. Nonetheless, when the response function presents local behavior such as spikes or step changes, although a good fit can be theoretically achieved by the PCE increasing the polynomial order, the corresponding computational cost to estimate the parameters can become burdensome.

C. KRIGING

Kriging is a stochastic interpolation algorithm, which assumes that the model output $Y = f(\mathbf{X})$ is the realization of a Gaussian process indexed by $\mathbf{X} \in D_X \subset \mathbb{R}^M$ where, in our case, D_X is the domain of validity of the meta-model and M is the dimensionality of input state space (Kleijnen, 2009; Matheron, 1963). In practice, Kriging is a linear regression model where the residuals are correlated by means of a Gaussian process, instead of being independent:

$$Y = f(\mathbf{X}) = N(h(\mathbf{X})^T \boldsymbol{\beta}, \sigma^2 Z(\mathbf{X})), \quad (25)$$

where $h(\mathbf{X})^T \boldsymbol{\beta}$ represents the mean value, also known as trend, which is a general linear regression model (e.g., $h(\mathbf{X})$ can involve polynomial terms and it reflects the prior knowledge about the model), σ^2 is

the variance of the Gaussian process and $\mathbf{Z}(\mathbf{X})$ is a zero mean, unit variance stationary Gaussian process whose underlying correlation function is represented by $R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$. The correlation function typically depends on the distance of the two vectors \mathbf{x}, \mathbf{x}' : the closer they are, the higher their correlation. Due to the Gaussian process hypothesis, every set of realizations of the model output can be described by a Gaussian vector:

$$\begin{bmatrix} \hat{Y}(\mathbf{x}) \\ \mathbf{y} \end{bmatrix} \sim N_{N_{Krig}+1} \left(\begin{bmatrix} h(\mathbf{x})^T \boldsymbol{\beta} \\ \mathbf{H} \boldsymbol{\beta} \end{bmatrix}; \sigma^2 \begin{bmatrix} 1 & \mathbf{r}^T(\mathbf{x}) \\ \mathbf{r}(\mathbf{x}) & \mathbf{R} \end{bmatrix} \right). \quad (26)$$

Assuming that $\mathbf{y} = (y_1, \dots, y_{N_{Krig}})$ is an experimental design with associated information matrix \mathbf{H} and correlation matrix \mathbf{R} (i.e., $\mathbf{R}_{ij} = R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}; \boldsymbol{\theta})$, $i, j = 1, \dots, N_{Krig}$), then the prediction of the output \hat{Y} for a given configuration \mathbf{x} is given by:

$$\hat{Y}(\mathbf{x}) | \mathbf{y}, \sigma^2, \boldsymbol{\theta} \sim N(\mu_{\hat{Y}}; \sigma_{\hat{Y}}^2), \quad (27)$$

where

$$\mu_{\hat{Y}}(\mathbf{x}) = h(\mathbf{x})^T \boldsymbol{\beta} + r(\mathbf{x})^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{H} \boldsymbol{\beta}), \quad (28)$$

$$\sigma_{\hat{Y}}^2(\mathbf{x}) = \sigma^2(1 - r(\mathbf{x})^T \mathbf{R}^{-1} r(\mathbf{x})) + (h(\mathbf{x})^T - r(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{H})(\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} (h(\mathbf{x})^T - r(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{H})^T \quad (29)$$

with the regression coefficients estimated by $\boldsymbol{\beta} = (\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y}$.

One of the main advantages of this formulation is that a confidence interval can be associated to each prediction $\hat{Y}(\mathbf{x})$. This can be used for assessing the accuracy and precision of the meta-model: the smaller the confidence interval, the more precise the model prediction for the corresponding configuration.