Integrated Deterministic and Probabilistic Safety Analysis for Safety Assessment of Nuclear Power Plants

Guest Editors: Francesco Di Maio, Enrico Zio, Curtis Smith, and Valentin Rychkov
Integrated Deterministic and Probabilistic Safety Analysis for Safety Assessment of Nuclear Power Plants
Integrated Deterministic and Probabilistic Safety Analysis for Safety Assessment of Nuclear Power Plants

Guest Editors: Francesco Di Maio, Enrico Zio, Curtis Smith, and Valentin Rychkov
## Contents

Volume 2015, Article ID 136940, 2 pages

**Improved Modelling and Assessment of the Performance of Firefighting Means in the Frame of a Fire PSA**, Martina Kloos and Joel Peschke  
Volume 2015, Article ID 238723, 10 pages

**A Flooding Induced Station Blackout Analysis for a Pressurized Water Reactor Using the RISMC Toolkit**, Diego Mandelli, Steven Prescott, Curtis Smith, Andrea Alfonsi, Cristian Rabiti, Joshua Cogliati, and Robert Kinoshita  
Volume 2015, Article ID 308163, 14 pages

**Scenario Grouping and Classification Methodology for Postprocessing of Data Generated by Integrated Deterministic-Probabilistic Safety Analysis**, Sergey Galushin and Pavel Kudinov  
Volume 2015, Article ID 278638, 13 pages

**An Enhanced Preventive Maintenance Optimization Model Based on a Three-Stage Failure Process**, Ruifeng Yang, Jianshe Kang, and Zhenya Quan  
Volume 2015, Article ID 193075, 13 pages

**Safety Assessment of Nuclear Power Plants for Liquefaction Consequences**, Tamás János Katona, Zoltán Bán, Erzsébet Győri, László Tóth, and András Mahler  
Volume 2015, Article ID 727291, 11 pages

**Probabilistic Dynamics for Integrated Analysis of Accident Sequences considering Uncertain Events**, Robertas Alzbutas  
Volume 2015, Article ID 892502, 18 pages

**Demonstration of Emulator-Based Bayesian Calibration of Safety Analysis Codes: Theory and Formulation**, Joseph P. Yurko, Jacopo Buongiorno, and Robert Youngblood  
Volume 2015, Article ID 839249, 17 pages

**Delayed Station Blackout Event and Nuclear Safety**, Andrija Volkanovski and Andrej Prošek  
Volume 2015, Article ID 192601, 9 pages

**Risk-Based Clustering for Near Misses Identification in Integrated Deterministic and Probabilistic Safety Analysis**, Francesco Di Maio, Matteo Vagnoli, and Enrico Zio  
Volume 2015, Article ID 693891, 29 pages
Editorial

Integrated Deterministic and Probabilistic Safety Analysis for Safety Assessment of Nuclear Power Plants

Francesco Di Maio, Enrico Zio, Curtis Smith, and Valentin Rychkov

1Energy Department, Politecnico di Milano, Via La Masa 34, 20126 Milano, Italy
2Chair on System Science and Energetic Challenge, Fondation EDF, Centrale Supélec, Chatenay-Malabry Cedex, 92295 Paris, France
3Idaho National Laboratory, Idaho Falls, ID 83415, USA
4EDF Re&D, 92141 Clamart Cedex, France

Correspondence should be addressed to Francesco Di Maio; francesco.dimaio@polimi.it

Received 28 June 2015; Accepted 6 July 2015

The present special issue provides an overview of the research in the field of Integrated Deterministic and Probabilistic Safety Assessment (IDPSA) of Nuclear Power Plants (NPPs).

Traditionally, safety regulation for NPPs design and operation has been based on Deterministic Safety Assessment (DSA) methods to verify criteria that assure plant safety in a number of postulated Design Basis Accident (DBA) scenarios. Referring to such criteria, it is also possible to identify those plant Structures, Systems, and Components (SSCs) and activities that are most important for safety within those postulated scenarios. Then, the design, operation, and maintenance of these “safety-related” SSCs and activities are controlled through regulatory requirements and supported by Probabilistic Safety Assessment (PSA).

On the other hand, compliance with the evolving regulatory requirements and the drive for enhancements in safety and economics calls for the development of innovative deterministic and probabilistic approaches of assessment for the existing NPPs and their life extension. In this respect, a medium-term challenge is to combine the use of deterministic and probabilistic methodologies in an IDPSA framework, which includes the tools and methods that couple probabilistic and deterministic approaches.

In this special issue, this coupling is shown to

(i) improve the computational approach of NPPs Thermal-Hydraulic (TH) models, thanks to the employment of metamodels for multiparameter and nonlinear modelling and for the throughout exploration of the space of combinatorial plant scenarios. Gaussian Process- (GP-) based surrogate models are proposed for incorporating new information from tests or operating experience into TH codes, for reducing the uncertainty in the output of the code and for dealing with the computational burden required for calibrating parameters of the TH codes used for the deterministic simulation of the NPPs response in accident scenarios,

(ii) treat aleatory (stochastic aspects of accident scenarios) and epistemic (model and parameters) uncertainties in a consistent manner. A Risk Informed Safety Margin Characterization (RISMC) approach is proposed for treating the uncertainties (e.g., stochastic for the safety system recovery times and epistemic for the model parameters) for a joint quantification of the frequency of core damage probability and of the safety margin for a relevant NPP accident,

(iii) identify and characterize a priori unknown vulnerable scenarios. Two approaches are proposed: a characterization of a failure domain based on scenario clustering by decision trees, with the goal of highlighting the influence of timing and order of events in the scenario grouping thereby obtained and a risk-based clustering method for the identification of Prime Implicants (PI) and of Near Misses (NM) scenarios, that is, minimum vulnerable combinations of failure events that lead the system to fault or quasi-fault states (a condition close to accident), respectively,
(iv) describe the time-dependent and spatial interactions between physical phenomena, equipment failures, safety and nonsafety systems interactions, control logic, and operator actions. An improved model to describe the time-dependent interactions between fire and firefighting by simultaneously accounting for stochastic and epistemic uncertainties by a Fire Dynamics Simulator (FDS) embedded into a Monte Carlo Dynamic Event Tree (MCDET) is proposed. Also, a combined modelling approach based on the Theory of Probabilistic Dynamics (TPD) for the dynamic reliability modelling and simulation of an accident under uncertain events is proposed for the assessment of the probability of a NPP containment failure due to hydrogen combustion,

(v) reduce the reliance on expert judgment and simplifying (or overly conservative) assumptions about interdependencies. A deterministic analysis is integrated with a probabilistic analysis for the estimation of a NPP core damage frequency, with as little as possible inclusion of expert judgment into the assumption adoption. Also, highlighting the effects of decoupling, at different level of interdependence, the analysis of a NPP liquefaction from a ground vibration motion analysis due to different expert judgment on their mutual interaction is given,

(vi) reduce the overall costs of the analysis and reduce the maintenance costs of the integrated models compared to separate deterministic and probabilistic models. The benefit of an explicit treatment of time-dependencies and priorities of defects/failure occurrences by integrated deterministic and probabilistic models is shown for an improved planning of the maintenance activities on NPP components.

As this journal aims to promote development in the area of nuclear sciences and technologies, it has been natural to present this special issue reporting on recent research in the field of IDPSA in support to the sustainability and safety of the nuclear option and the renaissance of nuclear technology in the world and especially in those countries where nuclear programs have not yet been developed.

Acknowledgments

Finally, we wish to point out that this special issue would have not been possible without the outstanding contributions of all the authors and reviewers. Our sincere professional appreciation and personal gratitude go to all these people.

Francesco Di Maio
Enrico Zio
Curtis Smith
Valentin Rychkov
An integrated deterministic and probabilistic safety analysis (IDPSA) was carried out to assess the performances of the firefighting means to be applied in a nuclear power plant. The tools used in the analysis are the code FDS (Fire Dynamics Simulator) for fire simulation and the tool MCDET (Monte Carlo Dynamic Event Tree) for handling epistemic and aleatory uncertainties. The combination of both tools allowed for an improved modelling of a fire interacting with firefighting means while epistemic uncertainties because lack of knowledge and aleatory uncertainties due to the stochastic aspects of the performances of the firefighting means are simultaneously taken into account. The MCDET-FDS simulations provided a huge spectrum of fire sequences each associated with a conditional occurrence probability at each point in time. These results were used to derive probabilities of damage states based on failure criteria considering high temperatures of safety related targets and critical exposure times. The influence of epistemic uncertainties on the resulting probabilities was quantified. The paper describes the steps of the IDPSA and presents a selection of results. Focus is laid on the consideration of epistemic and aleatory uncertainties. Insights and lessons learned from the analysis are discussed.

1. Introduction

IDPSA—frequently also called Dynamic PSA—can be regarded as a complementary analysis to the classical deterministic (DSA) and probabilistic (PSA) safety analyses [1, 2]. It makes extensive use of a deterministic dynamics code and applies advanced methods for an improved modeling and probabilistic assessment of complex systems with significant interactions between a process, hardware, software, firmware, and human actions [3]. An IDPSA is particularly suitable in the frame of a fire PSA, since sequences of a fire interacting with the means to be applied for firefighting can be realistically modelled while aleatory uncertainties due to the stochastic aspects of the performances of the firefighting means can be simultaneously taken into account. Besides aleatory uncertainties, epistemic uncertainties can be considered as well. They may refer to parameters of the applied deterministic dynamics code and to the reliability parameters used to quantify the stochastic performances of the firefighting means.

An appropriate tool to conduct an IDPSA is MCDET (Monte Carlo Dynamic Event Tree) which allows for performing Monte Carlo (MC) simulation, the Dynamic Event Tree (DET) approach or a combination of both [4, 5]. Since MCDET can in principal be coupled to any deterministic dynamics code, the open source and freely available code FDS (Fire Dynamics Simulator) from NIST [6] was selected to be applied for fire simulation. What makes MCDET particularly useful for a fire safety analysis is its Crew Module which allows for considering human actions such as those applied for firefighting as a time-dependent process [7, 8] which can interact with the process modelled by any dynamics code chosen to be combined with MCDET such as FDS.

In the past, MCDET was already applied to analyse and assess the plant behaviour during a station black-out scenario with power supply recovery [4]. In that application, MCDET
was combined with the code MELCOR (version 1.8.5, [9]) for integrated severe accident simulation. In another application, MCDET was coupled to the thermal-hydraulics code ATHLET (mod 2.0, [10]) to assess the emergency operating procedure "Secondary Side Bleed and Feed" [7]. This procedure is to be employed in a pressurized water reactor (PWR) to achieve the protection goal of steam generator injection after the loss of feed-water supply.

The fire event selected to be analysed was assumed to occur in a compartment of a German reference nuclear power plant (NPP). The main question to be answered by the IDPSA was whether the plant specific firefighting means to be applied in case of a fire are able to protect those structures, systems, and components (SSC) in the compartment which are important to nuclear safety. Therefore, the most important analysis result was the probability of safety related SSC to be damaged by the fire. The influence of epistemic uncertainties on the probability was quantified.

Section 2 of this paper gives an overview on the methods implemented in MCDET. It is explained how these methods can be used to treat the aleatory and epistemic uncertainties of an IDPSA and how the influences of both types of uncertainties can be quantified. Details on the considered fire event, the plant specific firefighting means and on the modelling assumptions can be found in Section 3. The steps of the analysis and a selection of results are described in Section 4. Conclusions and lessons learned are presented in Section 5.

2. Methods Implemented in MCDET

The tool MCDET allows for performing Monte Carlo (MC) simulation, the Dynamic Event Tree (DET) approach, or a combination of both. How these methods can be used to consider aleatory uncertainties and to quantify their influence on the results of a deterministic dynamics code is described in Section 2.1. The method to handle epistemic uncertainties in addition to aleatory uncertainties and to get a quantification of their influence is topic of Section 2.2.

2.1. Consideration of Aleatory Uncertainties. Coupled with a deterministic dynamics code such as the FDS code, the tool MCDET can perform Monte Carlo (MC) simulation, the Dynamic Event Tree (DET) approach, or a combination of both [4, 5].

The DET approach is quite useful, if rare events like, for instance, the failures of safety systems which generally occur with small probabilities have to be considered. The first tool presented in literature which applied the DET approach is DYLAN [11, 12]. Other tools using the DET approach are, for instance, ADS-IDAC [13, 14], SCAIS [15, 16], ADAPT [17], and RAVEN [18].

The simulation of a DET starts with the calculation of a sequence running from the initial event until the occurrence of the first event for which aleatory uncertainties are to be taken into account (e.g., success/failure of a safety system). When this happens, a branching point is generated meaning that the calculations of all branches (alternative situations) which may arise at the corresponding point in time are launched, even those of low probabilities. For instance, at the point in time, when a safety system is demanded, both successful and failed operations of the system are considered and the corresponding simulation processes are launched. Each time when another event subjected to aleatory uncertainty occurs during the calculation of a branch, another branching point is generated and the simulations of the new branches are launched.

With MCDET, a conditional occurrence probability is assigned to each branch constructed in the course of a DET simulation. Multiplication of the conditional probabilities of all branches which made up a whole sequence finally gives the sequence probability. The probabilities of all sequences of a DET in general sum up to 1. If a probabilistic cut-off criterion was applied, the sum is smaller than 1, because all sequences with a conditional probability less than a given threshold value are ignored.

The DET approach avoids repeated calculations of dynamic situations shared by different sequences. Except for the first (root) sequence, any other sequence is calculated only from the time on where a corresponding branching occurs. The past history of a sequence is given by the parent sequence from which the sequence branches off, then, by the parent sequence of the parent sequence and so on.

One drawback of the DET approach is that a continuous variable like the timing of an event (e.g., the failure of a passive component) has to be discretized, if it is subjected to aleatory uncertainty. A coarse discretization would provide less accurate results. A detailed time discretization would lead to an exponential explosion of the number of branches. The accuracy of results derived from a more or less detailed discretization is difficult to quantify. To overcome this difficulty, MCDET allows for applying a combination of MC simulation and the DET approach which can adequately handle the aleatory uncertainty of any discrete or continuous variables and provide output data appropriate for quantifying the accuracy of the results, for instance, in terms of confidence intervals.

With MCDET coupled to a dynamics code, each DET is constructed on condition of values each randomly sampled for a continuous aleatory variable. Each new set of values for the continuous aleatory variables contributes to the generation of another DET. Result of this method is a sample of individual DETs, each constructed from a distinct set of values sampled for the continuous aleatory variables. The sampling of values for the continuous aleatory variables is not performed a priori, that is, before the calculation of a DET is launched. It is performed when needed in the course of the calculation. In this way, it is possible to treat not only the influence of aleatory uncertainties on the dynamics as calculated by the code but also the influence of the dynamics on aleatory uncertainties and to consider, for instance, a higher failure rate of a component, if a high temperature seriously aggravates the condition of the component.

From the conditional probabilities assigned to each sequence and the corresponding curves of safety related output quantities calculated by the dynamics code, the post-processing modules of MCDET can calculate the conditional
DET-specific and the unconditional scenario-specific distributions of safety related quantities. These scenario-specific distributions are the means over the corresponding DET-specific distributions. The accuracy of the resulting mean distributions and probabilities can be quantified in terms of 90% or 95% confidence intervals.

Figure 1 comprises two schematical illustrations of the sample of DETs generated by MCDET. In Figure 1(a), each DET of the sample is represented in the time-event space with focus on the events subjected to aleatory uncertainty (e.g., failure-on-demand of the systems S1, S2, and S3, error of human actions HA1, or failure of a passive component PC). Timing and order of events might differ from DET to DET due to the influence of the different values sampled for MC simulation. Associated with each sequence of events is the process state at each point in time as calculated by the applied dynamics code and the corresponding conditional probability. In Figure 1(a), the state of a process variable $P$ and the corresponding probability are exemplarily considered at the end of problem time. The probabilities over the range of $P$ (e.g., from 0 to 10) obtained from all sequences of a DET constitute a distribution at each point in time (e.g., at the end of problem time as shown in Figure 1(a)). Figure 1(b) shows each DET in the time-state space where the focus is laid on the temporal evolution of the process variable $P$ for each sequence of event.

MCDET also allows for performing pure MC simulation to consider aleatory uncertainties of discrete or continuous variables. Regardless of whether MC simulation, the DET approach or a combination of both are applied, the probabilities of damage states (e.g., the probability of safety related SSC to be damaged by a fire) can be directly related to those process quantities of the dynamics code which are used to define failure criteria (e.g., high target temperatures and exposure times).

2.2. Consideration of Epistemic Uncertainties. Like with continuous aleatory uncertainties, the influence of epistemic uncertainties is considered by Monte Carlo (MC) simulation. In a first step, the values of the parameters subjected to epistemic uncertainty (epistemic variables) are sampled. Then, for each element of that epistemic sample, a sample of individual DETs is generated. Each DET is constructed from the values of the epistemic sample element combined with respective values sampled for the continuous aleatory variables.

The approach applied to quantify the influence of epistemic uncertainties at least needs two distinct DETs to be simulated per vector of the epistemic sample. If this condition is fulfilled, the simulation results can be used to quantify the overall influence of the epistemic uncertainties on a representative value $R$ of the resulting scenario specific
probability distribution. A useful representative value is the expected value of the probability distribution, especially if probabilities such as the probability of a damage state are to be provided as IDPSA results. These probabilities can be represented as expected values of appropriately chosen Bernoulli distributions. For instance, the probability \( P(X > x) \) of variable \( X \) to exceed the value \( x \) is the expected value \( E(B) \) of the Bernoulli variable \( B \) with \( B = 1 \), if \( X > x \) and \( B = 0 \), if \( X \leq x \).

The expected value \( R_{\text{Ep}} \) of the scenario-specific distribution of a variable \( Y \) (Section 2.1) per epistemic vector is the mean over the expected values \( R_{\text{Ep,Al}} \) of the DET-specific probability distributions of the respective epistemic vector (Formula (1)). \( R_{\text{Ep}} \) varies as a function of the epistemic variables \( \text{Ep} \), while \( R_{\text{Ep,Al}} \) varies as a function of both the epistemic variables \( \text{Ep} \) and the continuous aleatory variables \( \text{Al} \):

\[
R_{\text{Ep}} := E(Y \mid \text{Ep}) = E(R_{\text{Ep,Al}} \mid \text{Ep}),
\]

where \( E(\cdot \mid \text{Ep}) \) denotes the conditional expectation of a variable \( Y \) or \( R_{\text{Ep,Al}} \) as a function of the epistemic variables \( \text{Ep} \).

Formula (1) is true due to the following relationship:

\[
E(R_{\text{Ep,Al}} \mid \text{Ep}) := E(E(Y \mid \text{Ep}, \text{Al}) \mid \text{Ep}) = E(Y \mid \text{Ep}),
\]

where \( E(Y \mid \text{Ep}, \text{Al}) \) denotes the conditional expectation of variable \( Y \) as a function of the epistemic (Ep) and continuous aleatory variables (Al).

Formula (2) derives from the known equation for conditional expectations:

\[
E(E(Y \mid X)) = E(Y),
\]

where \( X \) and \( Y \) denote two variables, \( E(Y) \) is the expectation of \( Y \), and \( E(Y \mid X) \) the conditional expectation of \( Y \) as a function of \( X \).

A quantification of the epistemic uncertainty of the expected value \( R_{\text{Ep}} \) of the scenario-specific distribution can be obtained by estimating the expectation \( E(R_{\text{Ep}}) \) and the variance \( \text{Var}(R_{\text{Ep}}) \) of \( R_{\text{Ep}} \) and by using the estimators, for instance, to derive the parameters of a distribution supposed to be appropriate for \( R_{\text{Ep}} \). If \( R_{\text{Ep}} \) represents a probability, the Beta distribution might be an adequate distribution assumption.

The expectation \( E(R_{\text{Ep}}) \) can be estimated as the arithmetic mean over the expected values \( R_{\text{Ep,Al}} \) of the DET-specific probability distributions. This is based on the equation for conditional expectations (Formulae (I) and (3)):

\[
E(R_{\text{Ep}}) = E(E(R_{\text{Ep,Al}} \mid \text{Ep})) = E(R_{\text{Ep,Al}}).
\]

The variance \( \text{Var}(R_{\text{Ep}}) \) can be calculated from the following known equation:

\[
\text{Var}(R_{\text{Ep}}) = \text{Var}(E(R_{\text{Ep,Al}} \mid \text{Ep}))
\]

\[
= \text{Var}(R_{\text{Ep,Al}}) - E(\text{Var}(R_{\text{Ep,Al}} \mid \text{Ep})),
\]

where \( \text{Var}(\cdot) \) denotes the variance of a variable (\( R_{\text{Ep}} \) or \( R_{\text{Ep,Al}} = E(R_{\text{Ep,Al}} \mid \text{Ep}) \)) and \( E(\text{Var}(R_{\text{Ep,Al}} \mid \text{Ep})) \) is the expectation of the conditional variance of \( R_{\text{Ep,Al}} \) given \( \text{Ep} \).

The estimators of the mean \( E(R_{\text{Ep}}) \) and variance \( \text{Var}(R_{\text{Ep}}) \) can also be applied to calculate well-known inequations from statistics such as those of Chebychev (Formula (6)) or Cantelli (Formula (7)). These inequations can then be used to quantify the epistemic uncertainty of \( R_{\text{Ep}} \) in terms of conservative estimations, for instance, of a 95% interval or of the 5%- or 95%-quantiles:

\[
P\left(\left|R_{\text{Ep}} - E(R_{\text{Ep}})\right| \geq t \right) \leq \frac{\text{Var}(R_{\text{Ep}})}{t^2}, \quad t \geq 0 \tag{6}
\]

\[
P \left(R_{\text{Ep}} \leq E(R_{\text{Ep}}) - t \right) \leq \frac{\text{Var}(R_{\text{Ep}})}{\text{Var}(R_{\text{Ep}}) + t^2}, \quad t \geq 0. \tag{7}
\]

Another alternative to quantify the epistemic uncertainty of \( R_{\text{Ep}} \) is the calculation of two or one-sided (95%; 95%) tolerance limits [19]. The only requirement of this alternative is a minimum number of runs which account for the variations due to epistemic uncertainties [20]. For instance, at least 59 values for \( R_{\text{Ep}} \) must be available to quantify the upper one-sided (95%; 95%) tolerance limit.

3. Fire Event, Firefighting Means, and Modelling Assumptions

The fire event considered in the analysis and the assumptions of the corresponding FDS model are described in Section 3.1 of this Section. An overview on the plant specific firefighting means with emphasis on human actions and information on how these means were modelled are given in Section 3.2.

3.1. Fire Event and Modelling Assumptions. The fire was assumed to occur in a compartment of a NPP including cooling and filtering equipment for pump lubrication oil and electrical cables routed below the ceiling. Since these cables carry out safety related functions, one aim of the analysis was to find out whether these cables can be sufficiently protected against the fire by the plant specific firefighting means. It was assumed that malfunction of the oil-heating system designated to heat up the pump lubrication oil in the start-up phase of the NPP causes an ignition of the oil.

The dimensions of the compartment where the fire was supposed to start are about \( w \times l \times h = 8 \times 6.2 \times 6 \) m. Compartment walls are from concrete. The compartment is divided into a lower and an upper level by a steel platform at 2.4 m height. This is where the electrical oil heater is located and the fire was assumed to start (Figure 2). The steel platform can be reached by steel stairs. The three compartment doors lead to the lower level of the compartment. The lower level at the compartment. It was assumed that one of these doors might be (randomly) left in open position. The corresponding probability was considered as epistemic uncertainty (Table 1). The mechanical air exchange by an air intake and an exhaust vent was considered to be 800 m\(^3\)/h. The air inlet duct (violet in Figure 2) has one diffuser above the fire and one to the lower level. The outlet
duct (yellow in Figure 2) sucks air from the upper layer by two diffusors which can be closed by the fire damper. The fire damper at the exhaust vent was supposed to close after melting of a fusible link at 72°C. The probability of this mechanism to fail was considered as epistemic uncertainty (Table 1). If the outlet damper is closed, the mechanical air supply into the room was considered to be reduced to 400 m^3/h. This value was chosen to account for increased pressure losses, if the inlet air leaves the room via other leakages.

The fire simulation was performed by the Fire Dynamics Simulator (FDS) 6.0 [6]. FDS is a large-eddy simulation code for low-speed flows with emphasis on smoke and heat transport from fires. As input of FDS, the fire compartment was discretized in one mesh with a grid solution of 0.2 m in all three directions. The evolution of the fire depends on the leakage rate of the oil and was considered to be linear over all three directions. The characteristic time to reach 1 MW heat release rate was varied from 250 s to 700 s (Table 1). Due to the assumed fire, the electrical cables below the ceiling are exposed to hot smoke and radiation. The thermal penetration of the cable material was described by the model for thermally induced electrical failure (THIEF) implemented in FDS. The THIEF model predicts the temperature of the inner cable jacket under the assumption that the cable is a homogeneous cylinder with one-dimensional heat transfer. The thermal properties—conductivity, specific heat, and density—of the assumed cable are independent of the temperature. In reality, both the thermal conductivity and the specific heat of polymers are temperature-dependent. In the analysis, conductivity, specific heat, density, and the depth of the cable insulation were considered as uncertain parameters with relevant influence (Table 1).

3.2. Firefighting Means. If equipment and procedures work as intended, firefighting is a rather short process, because the compartment where the fire is assumed to occur is equipped with a fixed fire extinguishing system which suppresses the fire with a sufficiently large amount of water after actuation by the fire detection and alarm system. However, if the automatic actuation of the fixed fire extinguishing system fails, the firefighting process is complex and essentially depends on the manual firefighting means performed by the plant personnel in charge.

There are three states of the fire detection and alarm system which can be assumed as decisive for the manual firefighting means, namely, at least two detectors, only one detector or none of the detectors indicating an alarm signal to the control room. If at least two fire detectors send an alarm signal, the control room operator (shift leader) immediately instructs the shift fire patrol and the on-site fire brigade to inspect the compartment and to perform the necessary steps for fire suppression. If there is a signal by only one detector, the signal might be a faulty or spurious one (e.g., due to dust, steam, etc.). This is why the fire patrol trained for fighting incipient fires is instructed to inspect the fire compartment and to verify the fire. Suppose the fire patrol verifies the fire, the shift leader, who is immediately informed, calls the on-site fire brigade. In the mean-time, the fire patrol tries to suppress the fire either by a portable fire extinguisher or by manually actuating the fixed fire extinguishing system from outside the fire compartment. If none of the fire detectors sends an alarm, the detection of the fire depends on the shift patrol inspecting the compartment at a random time once during a shift.

The fire patrol usually is the first person who arrives at the fire compartment. His/her success of suppressing the fire with a portable fire extinguisher was assumed to depend on the local optical density \( D \) of the smoke at 3.20 m height (0.80 m above the level of the platform). For optical densities below \( D = 0.1 \, \text{m}^{-1} \), it was assumed that the fire patrol can detect the fire and start to suppress it by means of a portable fire extinguisher after a delay of 10 s. For \( 0.1 \, \text{m}^{-1} < D < 0.4 \, \text{m}^{-1} \), it was assumed that the delay time until the fire source is detected and the suppression can be started increases with the optical density. The delay time was assumed to be \( D \times 100 \) seconds. For \( D \geq 0.4 \, \text{m}^{-1} \), fire suppression with a portable fire extinguisher and without any personal protective equipment was supposed to be impossible due to reduced visibility and irritant smoke effects on eyes and breathing organs. The fire patrol does not wear personal protective equipment. The threshold value of 0.4 m\(^{-1}\) for the optical density was considered as epistemic uncertainty (Table 1). If fire suppression with a portable fire extinguisher is not possible, the fire patrol can try to manually actuate the fixed fire extinguishing system. If this does not work, the fire brigade has to extinguish the fire with their equipment.

Besides the reliability of the fire detection and alarm system and the performance of human actions, the success of firefighting mainly depends on the reliability of active fire barrier elements such as fire dampers or doors and of the fire extinguishing systems which can be manually actuated.

The Crew Module of MCDET was used to model and simulate the time-dependent process of the actions of the plant personnel in charge of firefighting. The model was constructed on the basis of documents from the reference NPP and walk-talk-throughs at locations relevant for firefighting. The aleatory uncertainties taken into account with regard to the performances of the crew members relate to the timings of rather simple actions to be applied for firefighting.
Table 1: Epistemic parameters and specified probability distributions.

<table>
<thead>
<tr>
<th>Epistemic parameter</th>
<th>Reference value</th>
<th>Distribution</th>
<th>Distribution parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value of optical density at which emerging smoke is visible</td>
<td>0.3</td>
<td>uniform</td>
<td>Min = 0.2, Max = 0.4</td>
</tr>
<tr>
<td>under fire compartment door [1/m]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response time index for activation temperature of fire dampers</td>
<td>125</td>
<td>uniform</td>
<td>Min = 50, Max = 200</td>
</tr>
<tr>
<td>[√m⋅s]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threshold of optical density (D) below which fire compartment can be entered [1/m]</td>
<td>0.4</td>
<td>uniform</td>
<td>Min = 0.3, Max = 0.5</td>
</tr>
<tr>
<td>Fraction of fuel mass (oil) converted into smoke</td>
<td>0.097</td>
<td>uniform</td>
<td>Min = 0.095, Max = 0.099</td>
</tr>
<tr>
<td>Time to reach 1 MW heat release rate [s]</td>
<td>425</td>
<td>uniform</td>
<td>Min = 250, Max = 700</td>
</tr>
<tr>
<td>Conductivity of cable [W/m*K]</td>
<td>0.275</td>
<td>uniform</td>
<td>Min = 0.15, Max = 0.4</td>
</tr>
<tr>
<td>Specific heat of cable [kJ/kg*K]</td>
<td>1.225</td>
<td>uniform</td>
<td>Min = 0.95, Max = 1.5</td>
</tr>
<tr>
<td>Depth of cable isolation [m]</td>
<td>0.0016</td>
<td>uniform</td>
<td>Min = 0.0012, Max = 0.002</td>
</tr>
<tr>
<td>Cable density [kg/m^3]</td>
<td>1131</td>
<td>uniform</td>
<td>Min = 833, Max = 1430</td>
</tr>
<tr>
<td>Specific heat of concrete [kJ/kg*K]</td>
<td>0.65</td>
<td>uniform</td>
<td>Min = 0.5, Max = 0.8</td>
</tr>
<tr>
<td>Conductivity of concrete [W/m*K]</td>
<td>1.75</td>
<td>uniform</td>
<td>Min = 1.4, Max = 2.1</td>
</tr>
<tr>
<td>Thickness of concrete walls in the fire compartment [m]</td>
<td>0.37</td>
<td>uniform</td>
<td>Min = 0.32, Max = 0.42</td>
</tr>
<tr>
<td>Probability that a fire door falsely stays open</td>
<td>0.005</td>
<td>beta</td>
<td>(\alpha = 1.5, \beta = 236.5)</td>
</tr>
<tr>
<td>Probability that fire damper fails to close</td>
<td>0.01</td>
<td>beta</td>
<td>(\alpha = 1.5, \beta = 117.5)</td>
</tr>
<tr>
<td>Failure temperature of I&amp;C cables [°C]</td>
<td>170</td>
<td>uniform</td>
<td>Min = 145, Max = 195</td>
</tr>
<tr>
<td>Critical time periods [s] with temperatures of I&amp;C cables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\geq 145^\circ C)</td>
<td>420</td>
<td>uniform</td>
<td>Min = 360, Max = 480</td>
</tr>
<tr>
<td>(\geq 150^\circ C)</td>
<td>300</td>
<td>uniform</td>
<td>Min = 240, Max = 360</td>
</tr>
<tr>
<td>(\geq 160^\circ C)</td>
<td>215</td>
<td>uniform</td>
<td>Min = 180, Max = 250</td>
</tr>
<tr>
<td>(\geq 170^\circ C)</td>
<td>160</td>
<td>uniform</td>
<td>Min = 120, Max = 200</td>
</tr>
<tr>
<td>(\geq 180^\circ C)</td>
<td>80</td>
<td>uniform</td>
<td>Min = 40, Max = 120</td>
</tr>
</tbody>
</table>

and to whether actions are successfully performed or not. A more detailed description of the model and the aleatory uncertainties can be found in [8].

4. Analysis Steps and Results

The analysis presented here was rather complex. An overview on the main analysis steps is given in Section 4.1 of this Section. A selection of results can be found in Section 4.2.

4.1. Analysis Steps. The first steps of the analysis focused on the stochastic performances of the crew members in charge of firefighting. The tool applied was just MCDET with its Crew Module. The corresponding simulations ran very fast and provided more than 100 DETs for each of several conditions. The conditions were identified as being decisive to the human actions applied for firefighting. They were given by the relevant states of the fire detection and alarm system (none, only one or at least two of the detectors operate as required) and by the fire progression (e.g., visibility of smoke in front of the door when shift personnel reaches the fire compartment or production of smoke in the fire compartment). Running MCDET and its Crew Module for a set of prescribed conditions related to the fire progression was necessary, since FDS was not applied in this part of the analysis.

From the DETs resulting from the simulations of MCDET only, various conditional distributions could be derived by using the corresponding postprocessing module of MCDET. The distributions refer to the timings of complex sequences of human actions such as the time period between fire alarm and the arrival of fire fighters at the fire compartment door or the time period between the arrival at the fire compartment door and the beginning of fire extinguishing. They express the stochastic variability of the timings and were used as input to the simulations performed in the second part of the analysis.

The second part of the analysis dealt with the modelling, simulation and evaluation of the interaction of the fire dynamics with relevant factors subjected to uncertainty and affecting the fire dynamics. The aleatory uncertainties taken into account refer to the timing and outcome (success/error) of human action related events, the operability of the fire detection and alarm system as well as to the functioning of active fire barrier elements (i.e., fire dampers and fire doors) and of the fire extinguishing systems which could be manually activated.

The main tools applied in the second part of the analysis were MCDET (without its Crew Module) and FDS. Modelling assumptions on the fire event were specified as input to FDS (Section 3.1) while the relevant parameters subjected
to aleatory uncertainty as well as the corresponding distributions and branching information quantifying the aleatory uncertainty (Section 2.1) were entered as input to MCDET. The parameters considered as potentially important and subjected to epistemic uncertainty (Table 1) were part of the MCDET input as well. The values of these parameters were sampled by the tool SUSA 3.6 for uncertainty and sensitivity analysis [21] and, then, provided as input to MCDET.

The simulations of FDS and those of MCDET were supervised by the old version of the MCDET Scheduler which allowed for calculating each DET in one process. The simulation approach made extensive use of the restart capabilities of the FDS code.

Fire sequences were planned to run up to 1800 s (0.5 hours) after ignition. If the fire suppression had started in a sequence, the simulation of that sequence was stopped as soon as the temperature inside the jacket of safety related cables fell below 120°C. It was assumed that the fire is under control in that situation and that a temperature below 120°C does not cause any harm to the cables.

The output of the simulations comprised data of about 2400 different fire sequences from a sample of 120 individual DETs. Two distinct DETs were simulated per vector of the epistemic sample (cf. Section 2.2). That means the epistemic sample included 60 different vectors. For the evaluation of the output data, corresponding postprocessing modules of MCDET were applied.

4.2. Analysis Results. The safety related targets which could be damaged and, therefore, were selected to be considered in the analysis, are I&C cables routed below the ceiling of the fire compartment (Section 3.1). The FDS output quantity used as indicator for cable damage is the temperature inside the jacket of the cables.

If the fixed fire extinguishing system of the compartment where the fire starts can be automatically actuated, the fire can be suppressed rather quickly without causing significant damage to the safety related targets (Section 3.2). The situation is more critical, if the automatic actuation of the fixed fire extinguishing system of the compartment fails. Therefore, the analysis focused on that condition. All results presented in the following refer exclusively to that condition. That means all probabilities presented are conditional probabilities.

Figure 3 shows the temporal evolution of the temperature inside the cable jacket for those sequences of all generated DETs where the fire detection and alarm system operates as required. Differences between the sequences are due to the overall influence of aleatory and epistemic uncertainties. Distinct colours used in Figure 3 indicate which sequences lead to successful fire suppression (green and red curves) and which not (black curves). Successful fire suppression can be performed either by the fire patrol (green curves) or by the fire brigade (red curves). The fire patrol can extinguish the fire by a portable fire extinguisher or by manually actuating the stationary fire extinguishing system from outside the fire compartment (Section 3.2). If the fire detection and alarm system operates as required, the fire patrol can suppress the fire mostly within 800 s (~13 min) after fire ignition. If the patrol fails to suppress the fire, the fire brigade can extinguish the fire with their equipment. The fire brigade succeeds to suppress the fire mostly within 800 to 1200 s (~13 to 20 min)—in some cases within 1200 to 1500 s (20 to 25 min)—after fire ignition.

For sequences without any fire extinguishing within 1800 s (black curves), three distinguished temperature clusters are clearly visible in Figure 3. In the first cluster, the temperature inside the cable jacket decreases below 100°C within 1800 s after ignition. The associated sequences are characterized by the corresponding fire damper operating as demanded and the fire door being closed (Section 3). The same is true for the sequences of the second cluster with a temperature on a level between 105°C and 120°C. The reasons...
for a smaller temperature decrease compared to that of the first cluster must be further investigated. In the third cluster, the temperature remains at a rather high level between 125 °C and 150 °C up to the end of simulation time (1800 s). This is the consequence of an open fire damper or an open fire door.

Figure 4 shows the overall distribution and two conditional distributions of the maximum temperature inside the cable jacket. The conditional distributions refer to the conditions, if the fire is successfully extinguished by the plant personnel or not. If the fire can be successfully suppressed, the maximum temperature can be reduced below 150 °C with a mean probability of about 0.87. If the fire cannot be suppressed, the mean probability of a maximum temperature below 150 °C is approx. 0.51. The main effect on the maximum temperature results from the actions of the fire patrol, because he/she can start the fire suppression quite early and therefore avoid a higher temperature maximum inside the cable jacket (cf. Figure 3). The overall distribution and the conditional distribution referring to fire suppression by the plant personnel are nearly identical. This is due to the very low conditional probability for failed fire suppression within 1800 s. The mean value is 5.97E-06.

Distributions referring to the time period with the temperature inside the cable jacket exceeding 160 °C are shown in Figure 5. They indicate, for instance, that the time period is not longer than approximately 113 s with a mean probability of 0.95, if the fire can be successfully suppressed. This result is also applicable to the overall unconditional distribution because of the very high conditional probability for successful fire suppression. If the fire cannot be suppressed within 1800 s, the 95%-quantile of the time period is 192 s. The differences between the two conditional time distributions in Figure 5 seem to be not very large. Nevertheless, these differences might be relevant, if the time period with a high temperature inside the cable jacket is used as criterion for cable failure.

Estimates of the probability of the I&C cables to be damaged by the fire were derived on the basis of two different failure criteria. According to failure criterion 1, the cables were assumed to be damaged, if the temperature inside the cable jacket exceeds a critical value. With criterion 2, the failure of a cable was supposed to be determined by both a high level of the temperature inside the cable jacket and a critical time period with the temperature being on a high level. Reference values and uncertainty quantifications referring to the failure temperature (criterion 1) and the critical time periods (criterion 2) are specified in Table 1. It is emphasized that the specifications are used only for demonstration purposes. They were derived from available experimental data on failure temperatures of I&C cables [22] and should be checked for real applications, in particular, with respect to the critical time periods for given temperature levels.

The mean probability $P$ of I&C cables to be damaged by the fire was estimated to be 1.76E-02 based on failure criterion 1 and 4.12E-03 based on criterion 2 (Table 2). Figure 6 shows, for each criterion, the cumulative distribution of the cable failure probability $P_{Ep}$ calculated per epistemic sample vector. As mentioned in Section 4.1, the epistemic sample included 60 elements. The two distributions of $P_{Ep}$ differ significantly.

According to the distribution related to the first criterion, the possible values for $P_{Ep}$ range from zero to 5.12E-01 and the (subjective) probability of $P_{Ep}$ to be zero—meaning that the critical temperature threshold is not exceeded—is relatively high. It can be concluded from the distribution related to the second criterion that even if the temperature always remains below the critical threshold, there are sequences with a temperature ranging on a relatively high level for a rather long time exceeding the critical time period. The probabilities for those sequences range up to 1.53E-02. Furthermore, it can be concluded, that if the temperature exceeds a critical
threshold (e.g., with a relatively high probability of 5.12E-01), it is often only for a short time considered not being critical according to the second criterion.

Since 60 values are available to quantify the epistemic uncertainty of the probability $P_{Ep}$, the one-sided upper (95%, 95%) tolerance limit could be calculated (see Section 3.2). Based on failure criterion 1, the (95%, 95%) tolerance limit according to Wilks formula [19] is 5.12E-01. The (95%, 95%) tolerance limit based on criterion 2 is 1.53E-02.

Estimates of the expectation $E(P_{Ep})$ and the variance $\text{Var}(P_{Ep})$ of $P_{Ep}$ as a function of the epistemic uncertainties are derived based on the formulae in Section 3.2. They are given in Table 2. Applying these estimates to the inequation of Cantelli (Formula (7)), the 95% quantile of $P_{Ep}$ is estimated to be 2.81E-01 based on criterion 1 and 3.00E-02 based on criterion 2. These estimates differ from the (95%, 95%) tolerance limits calculated according to Wilks formula. With regard to criterion 1, the tolerance limit is nearly two times higher, whereas it is two times smaller with regard to criterion 2.

If $E(P_{Ep})$ and $\text{Var}(P_{Ep})$ are the expectation and the variance of the Beta distribution assumed to quantify the epistemic uncertainty of $P_{Ep}$, the corresponding 95% quantile is given by 1.10E-01 according to criterion 1 and by 1.61 E-02 according to criterion 2. A comparison of the results from the applied approaches for epistemic uncertainty quantification underlines what was expected, namely, that the most conservative estimation of the epistemic uncertainty of $P_{Ep}$ is the tolerance limit (criterion 1) or the estimate of the 95% quantile derived from the inequation of Cantelli (criterion 2).

### Table 2: Estimates of the expectation $E(P_{Ep})$ and the variance $\text{Var}(P_{Ep})$ of the epistemic probability $P_{Ep}$.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Estimate of $E(P_{Ep})$</th>
<th>Estimate of $\text{Var}(P_{Ep})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Criterion 1</td>
<td>1.76E – 02</td>
<td>3.64E – 03</td>
</tr>
<tr>
<td>Criterion 2</td>
<td>4.12E – 03</td>
<td>3.55E – 05</td>
</tr>
</tbody>
</table>

5. Conclusions

An IDPSA was successfully performed to assess the performance of the firefighting means to be applied in a nuclear power plant. The application of advanced methods allowed for an improved modelling of the interaction between the fire and the firefighting means whilst simultaneously taking account of aleatory and epistemic uncertainties.

The analysis was performed in two parts. The first steps of the analysis focused on the performances of the crew members in charge of firefighting and made extensive use of the tool MCDET and its Crew module. The aleatory uncertainties taken into account relate to the timings of human actions to be applied for firefighting and to whether actions are successfully performed or not. The simulation results provided by MCDET and its Crew module were conditional distributions quantifying the aleatory uncertainties of the timings of complex sequences of human actions. These distributions were used as input to the subsequent analysis steps dealing with the fire dynamics interacting with the plant specific means designated to be applied for firefighting including the functions of fire extinguishing systems and active fire barriers. The aleatory uncertainties considered in that part of the analyses refer to the performances of these means. Epistemic uncertainties were taken into account as well and mainly relate to parameters of the code FDS applied for fire simulation.

From the huge amount of output data provided by the application of the coupling of the tools FDS and MCDET, many different distributions referring to the temporal evolution of the temperatures of safety related targets could be calculated. Main result was the probability of safety related SSC to be damaged by the fire. A realistic estimate of this probability was derived based on a failure criterion considering both critical target temperatures and exposure times. It may be used, for instance, in the subsequent steps of a fire safety analysis to assess the further consequences of the assumed fire.

The IDPSA performed with MCDET and FDS also allowed for quantifying the influence of epistemic uncertainties. Depending on the number of simulation runs which can be afforded to account for epistemic uncertainties, various quantification approaches could be applied. Appropriate approaches are the tolerance limits according to Wilks formula or the estimation of quantiles based on well-known inequations from statistics such as the inequations of Chebychev or Cantelli. The use of an inequation may be appropriate, if the minimum number of runs required for calculating tolerance limits cannot be afforded.

A standard PSA is not able to provide the kind of results which can be obtained by an MCDET analysis. It mainly relies on logical event tree/fault tree models which are static and, therefore, cannot adequately account for timing effects. Different from an MCDET analysis where the timing and order of stochastic events (e.g., with regard to firefighting means) are automatically calculated by a dynamics code (e.g., FDS) coupled to MCDET, a standard PSA requires the analyst to prescribe the chronological order of events. This may have the effect that potentially important sequences with another order of events are not considered at all, and therefore, the incompleteness uncertainty associated with the standard PSA model (which may be already high due to a deficient handling of timing effects) is further increased. Furthermore, in a standard PSA, the physical-chemical process (e.g., the fire evolution) is calculated just for a few selected sequences. For most of the sequences, the behaviour of the process is not known and must be estimated from the few available results. This makes it nearly impossible to consider complex interactions between a process and stochastic events, especially if the timing of events is important.

The main lesson learned from the MCDET analysis was the importance of having a well validated and tested dynamics model when performing an IDPSA. The amount of work necessary to make the model of the fire scenario implemented in FDS applicable with the combinations of input data provided by MCDET was higher than initially expected. Besides that, a lot of activities had to be spent to find a way of how to handle the limited restart capabilities of FDS in order to make the code running in combination with
MCDET while simultaneously avoiding excessive calculation time and data storage. Nevertheless, the coupling of MCDET and FDS which is now available can be used for further IDPSAs performed in the frame of a fire safety analysis.

**Conflict of Interests**

The authors declare that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

The authors wish to thank their GRS colleagues B. Forell and M. Röwekamp for their useful contributions to this work which was sponsored by the Ministry of Economics and Technology (BMWi) within the frame of the Research and Development Project RSI198.

**References**


Research Article

A Flooding Induced Station Blackout Analysis for a Pressurized Water Reactor Using the RISMC Toolkit

Diego Mandelli, Steven Prescott, Curtis Smith, Andrea Alfonsi, Cristian Rabiti, Joshua Cogliati, and Robert Kinoshita

Idaho National Laboratory (INL), 2525 Fremont Avenue, Idaho Falls, ID 83415, USA

Correspondence should be addressed to Diego Mandelli; diego.mandelli@inl.gov

Received 18 December 2014; Accepted 17 May 2015

Copyright © 2015 Diego Mandelli et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper we evaluate the impact of a power uprate on a pressurized water reactor (PWR) for a tsunami-induced flooding test case. This analysis is performed using the RISMC toolkit: the RELAP-7 and RAVEN codes. RELAP-7 is the new generation of system analysis codes that is responsible for simulating the thermal-hydraulic dynamics of PWR and boiling water reactor systems. RAVEN has two capabilities: to act as a controller of the RELAP-7 simulation (e.g., component/system activation) and to perform statistical analyses. In our case, the simulation of the flooding is performed by using an advanced smooth particle hydrodynamics code called NEUTRINO. The obtained results allow the user to investigate and quantify the impact of timing and sequencing of events on system safety. In addition, the impact of power uprate is determined in terms of both core damage probability and safety margins.

1. Introduction

The Risk-Informed Safety Margin Characterization (RISMC) Pathway develops and delivers approaches to manage safety margins [1]. This important information supports the nuclear power plant owner/operator decision-making associated with near- and long-term operation. The RISMC approach can optimize plant safety and performance by incorporating a novel interaction between probabilistic risk simulation and mechanistic codes for plant-level physics. The new functionality allows the risk simulation module to serve as a “scenario generator” that feeds information to the mechanistic codes. The effort fits with the goals of the RISMC Pathway, which are twofold:

(1) To develop and demonstrate a risk-assessment method coupled to safety margin quantification: the method can be used by decision-makers as part of their margin management strategies.

(2) To create an advanced RISMC toolkit: this RISMC toolkit would enable users to have a more accurate representation of nuclear power plant safety margins and its associated influences on operations and economics.

When evaluating the safety margin, what we want to understand is not just the frequency of an event like core damage but how close we are (or are not) to key safety-related events and how we might increase our safety margin through proper applications of Risk-Informed Margin Management (RIMM). In general terms, a “margin” is usually characterized in one of two ways:

(i) A deterministic margin, typically defined by the ratio (or, alternatively, the difference) of a capacity (i.e., strength) over the load.

(ii) A probabilistic margin, defined by the probability that the load exceeds the capacity.

A probabilistic safety margin is a numerical value quantifying the probability that a safety metric (e.g., for an important process observable such as clad temperature) will be exceeded under accident scenario conditions.
The RISMC Pathway uses the probabilistic margin approach to quantify impacts on reliability and safety. As part of the quantification, we use both probabilistic (via risk simulation) and mechanistic (via physics models) approaches, as represented in Figure 1. Safety margin and uncertainty quantification rely on plant physics (e.g., thermal-hydraulics and reactor kinetics) coupled with probabilistic risk simulation. The coupling takes place through the interchange of physical parameters (e.g., pressures and temperatures) and operational or accident scenarios.

2. The RISMC Toolkit

In order to perform advanced safety analysis, the RISMC project has a toolkit that was developed internally at INL using MOOSE [2] as the underlying numerical solver framework. This toolkit consists of the following software tools (see Figure 2):

(i) RELAP-7 [3] (see Section 2.1): it is the code responsible for simulating the thermal-hydraulic dynamics of the plant.

(ii) RAVEN [4] (see Section 2.2): it has two main functions: (1) act as a controller of the RELAP-7 simulation and (2) generate multiple scenarios (i.e., a sampler) by stochastically changing the order and/or timing of events.

(iii) PEACOCK [5] (see Section 2.3): it is the Graphical User Interface (GUI) that allows the user to create/modify input files of both RAVEN and RELAP-7. It monitors the simulation in real time while it is running.

(iv) GRIZZLY [6]: it is the code that simulates the thermal-mechanical behavior of components in order to model component aging and degradation. Note that, for the analysis described in this paper, aging was not considered.

This paper presents an analysis that evaluates the impacts of power uprates on a SBO event caused by external flooding. Due to the nature of the problem, the thermal-mechanical modeling needed to simulate component aging is not required. Thus, RELAP-7, RAVEN, and PEACOCK are being used. In this respect, Sections 2.1, 2.2, and 2.3 describe in more detail the components of the RISMC toolkit that are here employed: RELAP-7, RAVEN, and PEACOCK.

2.1. RELAP-7. The RELAP-7 code [3] is the new nuclear reactor system safety analysis codes being developed at the Idaho National Laboratory (INL). RELAP-7 is designed to be the main reactor system simulation toolkit for the RISMC Pathway of the Light Water Reactor Sustainability (LWRS) Program [7]. The RELAP-7 code development is
RELAP-7 uses the INL’s MOOSE (Multi-Physics Object-Oriented Simulation Environment) framework for solving computational engineering problems in a well-planned, managed, and coordinated way. This allows RELAP-7 development to focus strictly on systems analysis-type physical modeling and gives priority to retention and extension of RELAP5’s multidimensional system capabilities.

A real reactor system is very complex and may contain hundreds of different physical components. Therefore, it is impractical to preserve real geometry for the whole system. Instead, simplified thermal-hydraulic models are used to represent (via “nodalization”) the major physical components and describe major physical processes (such as fluid flow and heat transfer). There are three main types of components developed in RELAP-7: (1) one-dimensional (1D) components, (2) zero-dimensional (0D) components for setting a boundary, and (3) 0D components for connecting 1D components.

2.2. RAVEN. RAVEN (Risk Analysis and Virtual Control Environment) [4] is a software framework that acts as the control logic driver for the thermal-hydraulic code RELAP-7. RAVEN is also a multipurpose Probabilistic Risk Assessment (PRA) code that allows dispatching different functionalities. It is designed to derive and actuate the control logic required to simulate both plant control system and operator actions (guided procedures) and to perform both Monte Carlo sampling [8] of random distributed events and dynamic branching-type [9] based analysis.

RAVEN consists of two main software components:
(1) Simulation controller.
(2) Statistical framework.

The first RAVEN component acts as controller of the RELAP-7 simulation while simulation is running. This control action is performed by using two sets of variables [10]:

(i) Monitored variables: the set of observable parameters that are calculated at each calculation step by RELAP-7 (e.g., average clad temperature).
(ii) Controlled parameters: the set of controllable parameters that can be changed/updated at the beginning of each calculation step (e.g., status of a valve (open or closed) or pipe friction coefficient).

The manipulation of these two data sets is performed by two components of the RAVEN simulation controller (see Figure 3):

(i) RAVEN control logic: it is the actual system control logic of the simulation where, based on the status of the system (i.e., monitored variables), it updates the status/value of the controlled parameters.
(ii) RAVEN/RELAP-7 interface: it is in charge of updating and retrieving RELAP-7/MOOSE component variables according to the control logic.

A third set of variables, that is, auxiliary variables, allows the user to define simulation specific variables that may be needed to control the simulation. From a mathematical point of view, auxiliary variables are the ones that guarantee the system to be Markovian [11]; that is, the system status at time \( t = t + \Delta t \) can be numerically solved given only the system status at time \( t = \tilde{t} \).

The set of auxiliary variables also includes those that monitor the status of specific control logic set of components (e.g., diesel generators, AC buses) and simplify the construction of the overall control logic scheme of RAVEN.

The RAVEN statistical framework is a recent add-on of the RAVEN package that allows the user to perform...
generic statistical analysis. By statistical analysis we include the following:

(i) Sampling of codes: either stochastic (e.g., Monte Carlo [8] and Latin Hypercube Sampling (LHS) [12]) or deterministic (e.g., grid and Dynamic Event Tree (DET) [9]).

(ii) Generation of Reduced Order Models (ROMs) [13] also known as surrogate models or emulators.

(iii) Postprocessing of the sampled data and generation of statistical parameters (e.g., mean, variance, and covariance matrix).

Figure 4 shows a general overview of the elements that comprise the RAVEN statistical framework:

(i) Model: it represents the pipeline between input and output space. It comprises both codes (e.g., RELAP-7) and also ROMs.

(ii) Sampler: it is the driver for any specific sampling strategy (e.g., Monte Carlo, LHS, and DET).

(iii) Database: it is the data storing entity.

(iv) Postprocessing module: it is the module that performs statistical analyses and visualizes results.

2.3. PEACOCK. PEACOCK is the GUI front end for the RELAP-7 code and, in general, for any generic MOOSE based application. It is a PYTHON based software interface that allows the user to interface both offline and online with the RELAP-7 simulation. The user can, in fact, both create/modify the RAVEN/RELAP-7 input file (offline) and monitor the RAVEN/RELAP-7 simulation while it is running (online). A screenshot of PEACOCK is given in Figure 5.

In the offline mode, the user has available all the blocks and components needed to build the RAVEN/RELAP-7 input file such as

(i) RELAP-7 simulation and component parameters,

(ii) RAVEN variables: monitored, controlled, and auxiliary (see Section 2.2),

(iii) RAVEN/RELAP-7 simulation output information.

3. PWR SBO Case Study

The purpose of this case study is to show the capabilities of the RISMC workflow in order to evaluate the impacts of power uprates on a PWR system during a SBO initiating event. This assessment cannot be easily performed in a classical ET/FT based environment [14] due to the fact that its logical structures do not explicitly consider simulation elements.

We employ the RISMC toolkit (see Section 2). This toolkit mixes advanced simulation based tools with stochastic
3.1. PWR System. A PWR simplified model has been set up based on the parameters specified in the OECD main steam line break (MSLB) benchmark problem [16]. The reference design for the OECD MSLB benchmark problem is derived from the reactor geometry and operational data of the TMI-1 nuclear power plant (NPP), which is a 2772 MW two-loop pressurized water reactor (see the system scheme shown in Figure 6).

In order to simulate a SBO initiating event we need to consider also the following electrical systems (see Figure 7):

(i) Primary power grid line 500 KV (connected to the 500 KV switchyard).
(ii) Auxiliary power grid line 161 KV (connected to the 161 KV switchyard).
(iii) Set of 2 diesel generators (DGs), DG1 and DG2, and associated emergency buses.
(iv) Electrical buses: 4160 V (step-down voltage from the power grid and voltage of the electric converter connected to the DGs) and 480 V for actual reactor components (e.g., reactor cooling system).
(v) DC system which provides power to instrumentation and control components of the plant. It consists of these two subsystems:

(a) Battery charger and AC/DC converter if AC power is available.
(b) DC batteries: in case AC power is not available.
3.2. SBO Scenario. The scenario considered is a loss of offsite power (LOOP) initiating event caused by an earthquake followed by tsunami-induced flooding. Depending on the wave height, it causes water to enter into the air intake of the DGs and temporary disable the DGs themselves. In more detail, the scenario is the following (see Figure 8):

1. An external event (i.e., earthquake) causes a LOOP due to damage of both 500 KV and 161 KV lines; the reactor successfully scrams and, thus, the power generated in the core follows the characteristic exponential decay curve.

2. The DGs successfully start and emergency cooling to the core is provided by the Emergency Core Cooling System (ECCS).

3. A tsunami wave hits the plant causing flooding of the plant itself. Depending on its height, the wave causes the DGs to fail and may also flood the 161 KV switchyard. Hence, conditions of SBO are reached (4160 V and 480 V buses are not energized); all core cooling systems are subsequently offline (including the ECCS).

4. Without the ability to cool the reactor core, its temperature starts to rise.

5. In order to recover AC electric power on the 4160 V and 480 V buses, three strategies based on the Emergency Operating Procedures (EOPs) are followed:

   (i) A plant recovery team is assembled in order to recover one of the two DGs.
   (ii) The power grid owning company is working on the restoration of the primary 161 KV line.
   (iii) A second plant recovery team is also assembled to recover the 161 KV switchyard in case it got flooded.

6. Due to its lifetime limitation, the DC battery can be depleted. If this is the case, even if the DGs are repaired, DGs cannot be started. DCs power restoration (through spare batteries or emergency backup DC generators) is a necessary condition to restart the DGs.

(7) When the 4160 KV buses are energized (through the recovery of the DGs or 161 KV line), the auxiliary cooling system (i.e., ECCS) is able to cool the reactor core and, thus, core temperature decreases.

3.3. Stochastic Parameters. For the scope of this paper, the following parameters are uncertain:

   \( t_{\text{wave}} \): time at which the tsunami wave hit the plant.
   \( h \): tsunami wave height.
   \( t_{\text{DG,rec}} \): recovery time of the DGs.
   \( t_{\text{PG,rec}} \): recovery time of the 161 KV power grid.
   \( t_{\text{batt, fail}} \): failure time of the batteries (DC system) due to depletion.
   \( t_{\text{batt, rec}} \): recovery time of the batteries (DC system).

For each of these parameters we will find the appropriate probability distribution function (see Section 4.3) in order to evaluate core damage probability \( P_{\text{CD}} \). Core damage is reached when max clad temperature in the core reaches its failure temperature (2200 F).

4. Case Study Modeling

This section shows how this PWR SBO analysis is being performed using the RISMC toolkit described in Section 2. In this respect, Figure 9 summarizes all the steps followed in this paper using the RISMC approach:

1. Initiating event modeling: it includes modeling characteristic parameters and associated probabilistic distributions of the event considered.
2. Plant response modeling: it includes modeling of the plant system dynamics.
(3) Components failure modeling: it includes modeling of specific components/systems that may stochastically change status (e.g., fail to perform specific actions) due to the initiating event or other external/internal causes.

(4) Scenario simulation: when all modeling aspects are complete (see previous steps), a set of simulations can be run by stochastically sampling the set of uncertain parameters.

(5) Given the simulation runs generated in Step 4, a set of statistical information (e.g., CD probability) is generated. We are also interested in determining the limit surface: the boundaries in the input space between failure and success.

4.1. Flood Modeling. A generic 3D facility model (see Figure 10) with conditions similar to the Fukushima incident was created and used to simulate various tsunami flooding examples. For initial testing only a slice of the entire facility (containing just a single unit) was used; this includes

- (i) turbine building,
- (ii) reactor building,
- (iii) offsite power facilities and switchyard,
- (iv) diesel generator (DG) building.

The 3D model is used as the collision geometry for any simulations. For this demonstration all objects are fixed rigid bodies; future analysis will explore the possibility of moving debris (caused by the flood) and possible secondary impacts due to this debris.

To mimic a tsunami entering the facility, a bounding container was added around the perimeter of the model and for the ocean floor. Then, over 12 million simulated fluid particles were added for the ocean volume. A wave simulator mechanism was constructed by having a flat planar surface that moves forward and rotates, pushing the water and creating a wave in the fluid particles.

Various wave heights can be generated by minor parameter adjustments to the movement of the wave generator. As the fluid particles are initially forced forward their movement energy is transferred and affects the particles around them using the mathematical equations for fluid physics built into the fluid solver.

There are many different approaches for simulating and optimizing fluid movement, each having different advantages and purposes. To achieve realistic and accurate results, a smooth particle hydrodynamics (SPH) based solver called NEUTRINO was used [17]. NEUTRINO also factors in advanced boundary handling and adaptive time stepping to help to increase accuracy and calculation speed. Most simulations were done using a sequential approach via 14
treads on seven cores at 2.4 Ghz and took approximately 3 minutes per frame with a total run time ranging from 75 to 90 hours depending on how many frames were needed for the simulation. With future code development, simulation time could be vastly improved by using distributed processing on computer cluster or coprocessor hardware.

As the particles of a simulation move, they interact with the rigid bodies of the 3D model. The simulated fluid flows around buildings, splashes, and interacts in a similar manner to real water. Measuring tools can also be added to the simulation to determine fluid contact information, water height, and even flow rates into openings at any given time in the simulation. This information can be used in two ways, a static success or failure depending on wave height, or a dynamic result based on time could be used for more detailed analysis.

Several simulations were run at different wave heights. The fluid penetration into the site is measured for each of the simulations to determine at what height the different systems fail. For our specific case, we are monitoring the venting for the DGs and the offsite power structures.

As shown in Figure 11 the fluid particles are penetrating both air intake vents for an 18 m wave. In more detail we know that at simulation time (or frame) 1275 DG1 fails from splash particles and DG2 fails at 1375.

4.2. Plant Mechanistic Modeling. The reactor vessel model consists of the Downcomers, the Lower Plenum, the Reactor Core Model, and the Upper Plenum. Three core channels (components with a flow channel and a heating structure) were used to describe the reactor core. Each core channel is representative of a region of the core (from one to thousands of real cooling channels and fuel rods).

In this analysis, the core model consists of three parallel core channels (hot, medium, and cold) and one bypass flow channel. Respectively, they represent the inner and hottest zone, the mid, and the outer and colder zone of the core. The Lower Plenum and Upper Plenum are modeled with branch models.

There are two primary loops in this model: Loop A and Loop B. Each loop consists of the Hot Leg, a heat exchanger and its secondary side pipes, the Cold Leg, and a primary Pump. A pressurizer is attached to the Loop A piping system to control the system pressure. Since a complex pressurizer model has not been implemented yet in the current version of RELAP-7 code, a Time Dependent Volume (pressure boundary conditions) has been used instead.

Figure 12 shows the core layout of the PWR model. The core height is 3.6576 m. The reactor consists of 177 fuel assemblies subdivided into 3 zones. The 45 assemblies in zone 1 are represented by the hot core channel and the 60 assemblies in zone 2 and 72 assemblies in zone 3 are, respectively, represented by the average core channel and the cold core channel (see Figure 13). The fuel assembly geometry data is taken from [16]. The reactor is assumed to be at end of cycle (EOC), 650 EFPD (24.58 GWD/THMt average core exposure), with a boron concentration of 5 ppm, and Xe and Sm at the equilibrium. The 3D core neutronics calculation results for the hot full power conditions are presented in [16].

Figure 13 shows the relative assembly radial power distribution for a quarter of the core. Using the values presented in Figure 13, the power distribution fraction and power density for each core channel are calculated and shown in Table 1. The power density is used as input to the RELAP-7 model to calculate the heat source.

4.3. Plant and Flooding Probabilistic Modeling. While Section 3.3 lists all the uncertainty parameters that are considered, this section focuses on the choice of probability distribution functions (pdfs) associated with these parameters.
Figure 12: Screenshot of the PWR model of RELAP-7 using PEACOCK.

Figure 13: Core zone correspondence (a) and assembly relative power (b) [3].
Regarding the time at which the tsunami wave hits the plant (i.e., \( t_{wave} \)), we were not able to obtain a representative distribution. Such time is equal to the distance of the epicenter of the earthquake that generated the tsunami wave divided by the average speed of the wave itself. Given the absence of this information, we chose to represent the uncertainty associated with \( t_{wave} \) as a uniform distribution defined between 0 and 4 hours. Thus we expected that a representative simulated wave would hit the plant site within 4 hours.

Regarding the DG recovery time \( (t_{DG,rec}) \), we used as a reference: the NUREG/CR-6890 vol. 1 [18]. This document uses a Weibull distribution with \( \alpha = 6.14 \) h and \( \beta = 3.8 \) h. This distribution represents the pdf of repair of one of the two DGs (choosing the one easiest to repair).

For the PG recovery time \( t_{PG,rec} \) we used as reference NUREG/CR-6890 vol.2 [19] (data collection was performed between 1986 and 2004). Given the four possible LOOP categories (plant centered, switchyard centered, grid related, or weather related), severe/extreme events (such as earthquake) are assumed to be similar to these events found in the weather category (these are typically long-term types of recoveries). This category is represented with a lognormal distribution (from NUREG/CR-6890) with \( \mu = 0.793 \) and \( \sigma = 1.982 \).

For the probability distribution for the wave height \( (h) \) we referred to [20] where an exponential distribution is defined. The average value of lambda (the characteristic parameter of the exponential distribution) is a function of return period (see Figure 14). The return period indicated the time span (in years) considered in the analysis. Figure 15 shows both probability and cumulative distribution functions (pdf and cdf) of wave heights \( h \) for three values of return periods (1, 10, and 100 years). For the scope of this paper, we assume a power uprate in conjunction with a 20-year life extension; thus, for a return period of 20 years we calculated a mean value of lambda equal to 0.206 m\(^{-1}\) (see Figure 14).

Regarding battery life (i.e., \( t_{batt,faul} \)), we chose to limit battery life between 4 and 6 hours using a triangular distribution. On the other hand, regarding the recovery time of the batteries (\( t_{batt,rec} \)), we used the method shown in [15] to model the pdf of human related actions. In [15], for human actions we looked into the SPAR-H [21] model contained in SAPHIRE. SPAR-H characterizes each operator action through eight parameters; for this study we focused on the two most important factors:

(i) Stress/stressors level.
(ii) Task complexity.

These two parameters are used to compute the probability that an action will happen or not; the probability values are then inserted into the event trees that contain such events. However, from a simulation point of view we are not seeking if an action is performed but rather when such action is performed. Thus, we need a probability distribution function that defines the probability that an action will occur as a function of time.

Since modeling of human actions is often performed using lognormal distributions [15] we chose a distribution where its characteristic parameters (i.e., \( \mu \) and \( \sigma \)) are dependent on the two factors listed above (stress/stressors level and task complexity). We used Table 2 [15] to convert the three possible values of the two factors into numerical values for \( \mu \) and \( \sigma \).

For the specific case of DC battery system restoration we assumed that the task has high complexity with extreme stress/stressors level. This leads to \( \mu = 45 \) min and \( \sigma = 15 \) min.

As part of the analysis we consider that the initiating event, that is, the tsunami wave, affects both the sequence of events and the probabilities associated with those events (see Figure 16). In particular, Figure 16 summarizes how wave height affects system dynamics by using a simplified event-tree structure:

(i) Wave height and DGs loss: DGs are intact and functional if the wave does not reach the exhaust inlet.

(ii) Wave height and recovery time of PG \( (t_{PG,rec}) \): the PG recovery time starts after the wave hits the plant. However, if the wave is high enough to reach the PG switchyard causing flooding on the switchyard itself then PG recovery time distribution \( t_{PG,rec} \) is changed. This change reflects the fact that more time is needed to clear/repair the switchyard facility. For our case the distribution of \( t_{PG,rec} \) is still lognormal but with a doubled mean value.

In conclusion, Table 3 summarizes the distribution associated with each uncertainty parameter.

### Table 2: Correspondence table between complexity and stress/stressors level and time values.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>( \mu ) (min)</th>
<th>Stress/stressors</th>
<th>( \sigma ) (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>45</td>
<td>Extreme</td>
<td>30</td>
</tr>
<tr>
<td>Moderate</td>
<td>15</td>
<td>High</td>
<td>15</td>
</tr>
<tr>
<td>Nominal</td>
<td>5</td>
<td>Nominal</td>
<td>5</td>
</tr>
</tbody>
</table>

![Figure 14: Mean value of lambda as a function of return period.](image-url)
2. Impact of Wave Height on DG and PG Status. We performed a series of simulations using the NEUTRINO code on the 3D plant model in order to measure plant response for several wave heights (see Section 4.2) in the interval range of [0 30] meters. The basic idea is to build a response function that can be implemented in the RAVEN control logic that, depending on the sampled parameter $h$ (wave height), it determines the status of both DGs and PG switchyard.

We found that the DGs tended to fail with smaller waves than the PG structures because the DG building is closer to the ocean shore and air intake vents face the wave directly (see Figure 17). In fact, if the wave is greater than 18 m, water enters in both DGs air intake while PG switchyard is flooded only for wave height greater than 30 m (see Table 4).

Note that, given the fact that the 3D plant model represents only a partial slice of the site and there is only a small

5. Safety Margin Analysis

This section presents in detail the series of results obtained by using the flooding simulation code NEUTRINO and the RAVEN/RELAP-7 plant response code. We focus our attention to

(i) evaluate the impact of wave height on plant response (see Section 5.1),

(ii) evaluate impact of power uprates on AC recovery timing (see Section 5.2),

(iii) evaluate impact of power uprates on CD probability (see Section 5.3).

### Table 3: Probability distribution functions for sets of uncertainty parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{wave}}$ (h)</td>
<td>Uniform [0.0, 4.0]</td>
</tr>
<tr>
<td>$t_{DG,\text{rec}}$ (h)</td>
<td>Weibull (alpha = 0.745, beta = 6.14)</td>
</tr>
<tr>
<td>$t_{PG,\text{rec}}$ (h)</td>
<td>Lognormal (mu = 0.793, sigma = 1.982)</td>
</tr>
<tr>
<td>$t_{PG,\text{rec}}$ (h)</td>
<td>Lognormal (mu = 1.586, sigma = 1.982)</td>
</tr>
<tr>
<td>$t_{\text{batt,fail}}$ (h)</td>
<td>Triangular (4.0, 5.0, 6.0)</td>
</tr>
<tr>
<td>$t_{\text{batt,rec}}$ (h)</td>
<td>Lognormal (mu = 0.75, sigma = 0.25)</td>
</tr>
<tr>
<td>$h$ (m)</td>
<td>Exponential (lambda = 0.206)</td>
</tr>
</tbody>
</table>

*a* If switchyard is not flooded by the wave.

*b* If switchyard is flooded by the wave.
opening to the backside of the facility that allows water to reach the PG switchyard, the PG switchyard may fail with smaller waves if a more complete model would be used.

5.2. Impact of Power Uprate on AC Recovery Time. As a second step, we started to evaluate how power uprates change the time to reach CD for different values of DG failure time. Two facts need to be considered:

(i) A power uprate implies that a higher energy is generated within the core and, hence, clad failure temperature is reached sooner.

(ii) A late DG failure time allows the ECCS to successfully remove more heat from the RPV. Since the decay heat curve is exponential we expect that such dependency is not linear.

This reduction in time to reach CD ranges from 3200 s to 4000 s; hence, on average the core reaches CD about an hour quicker if power level increases from 100% to 120%.

5.3. Probabilistic Analysis. While the analysis contained in Section 5.2 deterministically measures timing reduction due to power uprate, it does not show how such uprate probabilistically changes the probability to reach CD. In other words, how does an average time reduction of one hour to reach CD modify the actual probability of the CD event itself?

By using Latin Hypercube Sampling (LHS) available within the RAVEN statistical framework, we

(i) sampled $N$ times the distribution of the uncertain parameters listed in Table 3,

(ii) ran $N$ times RAVEN/RELAP-7 simulations with simulation parameter values changed accordingly to the sample values (generated in Step 1),

(iii) evaluated the overall CD probability by looking at the outcome of each RAVEN/RELAP-7 simulation.

Using the RAVEN statistical framework (see Section 2.2) we performed LHS of the distributions associated with the uncertain parameters listed in Table 3. We performed this sampling for both power levels: 100% and 120%. We then divided all the simulated scenarios (10,000 simulations for each power level) into four groups according to the ET structure shown in Figure 16.

From the obtained results, which are shown in Table 5, we can note the following:

(i) Probability of core damage $P_{CD}$ (branch 4 of Figure 16) increases from $217.8 \times 10^{-6}$ to $522.2 \times 10^{-6}$, an increase of 76%. Thus, the change in probability is

$$
\Delta P_{CD} = 304.417 \times 10^{-6}.
$$

(ii) Probability value associated with branch 1 (wave height does not disable DGs and, hence, AC power is always available throughout the simulation) since this value depends only on the wave height (i.e., if $\ell$ is less than 18 m).

A different way to view the $\Delta P_{CD}$ is to evaluate the limit surface [22] of the system: the boundaries in the input space ($\Omega$) between failure region ($\Omega^F$) and success region $\Omega^S$. For our cases, $\Omega = \Omega^F \cup \Omega^S$.

Obviously these boundaries are deterministically determined but probabilistic information can be generated by evaluating the CD probability as

$$
P_{CD} = \int_{\Omega^F} p(\omega) d\omega,
$$

<table>
<thead>
<tr>
<th>Branch</th>
<th>Outcome</th>
<th>Counter 100%</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OK</td>
<td>3657</td>
<td>0.9740</td>
</tr>
</tbody>
</table>
| 2      | OK      | 2764         | 18.3210^{-3}
| 3      | OK      | 2403         | 7.49810^{-3}
| 4      | CD      | 1176         | 217.810^{-6}
| 100%   |         |              |             |
| 1      | OK      | 3657         | 0.9740      |
| 2      | OK      | 2764         | 18.3210^{-3}
| 3      | OK      | 2403         | 7.49810^{-3}
| 4      | CD      | 1176         | 217.810^{-6}
| 120%   |         |              |             |
where \( p(\omega) d\omega \) is the probability associated with the volume \( d\omega \) of the input space.

In our applications, this integral is calculated using the stochastic sampling capabilities available in the RAVEN statistical framework.

Figure 18 shows the limit surface obtained in a two-dimensional input space, that is, DG failure time versus AC recovery time, for the two different cases: 100\% and 120\% power. From the stochastic samples we generated the limit surface using Support Vector Machines (SVMs) [23, 24].

When power increases it is expected that the failure region (red area) grows in the input space and, thus, also the probability of CD increases.

The value of \( \Delta P_{CD} \) is simply

\[
\Delta P_{CD} = \int_{\Omega_{120}^F - \Omega_{100}^F} p(\omega) d\omega, \tag{3}
\]

where \( \Omega_{120}^F \) and \( \Omega_{100}^F \) are the failure regions for the 120\% and 100\% power values.

6. Conclusions

In this paper we have summarized the series of steps that are needed to evaluate a RISMC detailed demonstration case study for an emergent issue using RAVEN and RELAP-7. We studied the impacts of power uprates on a flooding induced SBO event using the RISMC toolkit. We started by modeling both the PWR system dynamics using the RELAP-7 code and the flooding scenario using the NEUTRINO code.

Even though the RELAP-7 and NEUTRINO codes were not tightly coupled to each other (i.e., the flooding analysis causes triggers such as a DG failure that is captured in the RELAP-7 calculation), it was possible to evaluate the overall system response on a much greater level of detail than compared to classical ET/FT based methodologies.

Our statistical analysis was performed using the RAVEN code which allowed us to evaluate the impacts of power uprates on the overall probability of core damage. We also determined how plant recovery procedures get reduced in time due to the power uprate itself.

In this paper we particularly focused on steps that are necessary to complete such statistical analysis and the information that can be generated from it. This information can be used to perform decision-making for the three possible scenarios:

(i) Power uprate is feasible since core damage probability increase \( \Delta P_{CD} \) is below the acceptable limits.

(ii) Power uprate is not feasible since core damage probability increase \( \Delta P_{CD} \) is above the acceptable limits.

(iii) Even though \( \Delta P_{CD} \) is above the acceptable limits, power uprate is feasible if recovery procedures are enhanced.

For the third scenario, recovery procedure enhancement may include the following:

(i) Increase the height of the wave protection wall in order to reduce flooding level in the plant. This will act on the fraction of the wave height distribution that causes DG failure.

(ii) Improve AC emergency recovery procedures (e.g., FLEX system). This action acts directly on either the DG or PG recovery distribution (\( t_{DG_{rec}} \) and \( t_{PG_{rec}} \)), that is, a lower DG or PG average recovery time.

(iii) Move the DGs to a non-flood-prone area of the plant site.

(iv) Improve the bunkering of the DG building in order to reduce the likelihood of flood-caused failures.
Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


Research Article

Scenario Grouping and Classification Methodology for Postprocessing of Data Generated by Integrated Deterministic-Probabilistic Safety Analysis

Sergey Galushin and Pavel Kudinov

KTH, Division of Nuclear Power Safety, AlbaNova University Center, 106 91 Stockholm, Sweden

Correspondence should be addressed to Sergey Galushin; galushin@kth.se

Received 28 January 2015; Revised 27 March 2015; Accepted 31 March 2015

Academic Editor: Francesco Di Maio

Copyright © 2015 S. Galushin and P. Kudinov. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Integrated Deterministic-Probabilistic Safety Assessment (IDPSA) combines deterministic model of a nuclear power plant with a method for exploration of the uncertainty space. Huge amount of data is generated in the process of such exploration. It is very difficult to “manually” process and extract from such data information that can be used by a decision maker for risk-informed characterization, understanding, and eventually decision making on improvement of the system safety and performance. Such understanding requires an approach for interpretation, grouping of similar scenario evolutions, and classification of the principal characteristics of the events that contribute to the risk. In this work, we develop an approach for classification and characterization of failure domains. The method is based on scenario grouping, clustering, and application of decision trees for characterization of the influence of timing and order of events. We demonstrate how the proposed approach is used to classify scenarios that are amenable to treatment with Boolean logic in classical Probabilistic Safety Assessment (PSA) from those where timing and order of events determine process evolution and eventually violation of safety criteria. The efficiency of the approach has been verified with application to the SARNET benchmark exercise on the effectiveness of hydrogen management in the containment.

1. Introduction

Development of Deterministic Safety Analysis (DSA) and Probabilistic Safety Analysis (PSA) was crucial step for establishing state-of-the-art in nuclear power safety design and licensing. However, in order to avoid stagnation, it is important to recognize inherent limitations of the classical approaches and new opportunities provided by the overall progress of risk analysis science and computational technologies. For instance, advantage of DSA is that it can model dynamics of the plant systems driven by physical phenomena and their response to failures of the equipment or operator actions. If the “worst” scenarios can be clearly identified, then conservative treatment of uncertainties in DSA can be employed to estimate safety margins. The number of scenarios considered in DSA is usually small with respect to the actual set of possible accident scenarios, thus outcomes of DSA are largely affected by the expert judgment. However, obtaining a priori knowledge about “worst” case scenarios and “conservative” assumptions about uncertain parameters for complex systems is not a trivial task. PSA attempts to cover all possible risk significant scenarios. However, it is not easy to model a priori unknown dependency of the accident scenario outcome on the order and timing of the events (e.g., due to temporary evolution of the system parameters driven by complex physical processes and interactions) using Boolean logic of the classical PSA where the result is unambiguously determined by simple set of events. A robust safety justification must be based on both deterministic and probabilistic considerations to address the effects of the dynamic nature of mutual interactions between (i) stochastic disturbances (e.g., failures of the equipment), (ii) deterministic response of the plant (i.e., transients), (iii) control logic, and (iv) operator actions. Passive safety systems, severe accident, and
containment phenomena are examples of the cases when such dependencies of the accident progression on timing and order of events are especially important. Integrated use of deterministic and probabilistic safety analysis is a means to enable risk-informed decision making based on consistent evaluation of both the uncertainties arising from the stochastic nature of events (aleatory uncertainties) and those arising from lack of knowledge about the processes relevant to the system (epistemic uncertainties) [1].

Integrated Deterministic-Probabilistic Safety Assessment (IDPSA) methodologies aim to achieve completeness and consistency of the analysis through systematic consideration of different sources of uncertainties including physical processes, failures of hardware and software, and human actions. IDPSA tools usually employ (i) system simulation codes and models with explicit consideration of the effect of timing on the interactions between epistemic (modeling) and aleatory (scenario) uncertainties, (ii) a method for exploration of the uncertainty space. A review of the IDPSA methods for nuclear power plant applications can be found in [2].

For decision making, however, it is often insufficient to merely calculate a quantitative measure for the risk and respective uncertainties [3]. Detailed exploration of the uncertainty space usually results in huge amount of the data generated by the deterministic codes [4]. Therefore, one of the main problems for application of IDPSA methods is data post-processing and communication of the analysis results. Extracted information should be suitable for decision making and risk-informed characterization and eventually improvement of safety and performance of the system. Such understanding requires an approach to the interpretation, grouping of similar scenarios, and classification of the principal characteristics of the events that contribute to the risk. Several attempts to solve this problem has been undertaken. Different approaches have been developed to transient identification based on pattern classification by fuzzy C-means clustering [5], identification and classification of dynamic event tree scenarios via possibilistic clustering [6], probabilistic clustering for scenario analysis [7]. These methods use clustering tools and pattern recognition to identify and group similar scenarios that lead to failure.

The goal of this work is to develop methods that will enable understanding of the outcomes of IDPSA analysis while maintaining completeness. In order to achieve that, the methods should reduce the volume of the data generated by IDPSA tools without loss of important for decision making information. The strategy for the reduction of the data volume is based on (i) grouping of different scenarios into different “classes” according to different failure modes; (ii) identification of the scenarios that have “similar” behavior (clustering) within each class. Condensed information should provide useful insights into the complex accident progression and understanding of possible mitigation strategies.

In this work we develop an approach for classification and characterization of failure domains. Failure domain is a domain in the space of uncertain parameters where critical system parameters exceed safety thresholds. The approach is based on scenario grouping and clustering with application of decision trees for characterization of the influence of timing and order of the events. In this approach decision trees are constructed to represent failure domain as a set of leaf nodes and correspondent classification rules that lead to each node. The approach was applied to classification of the simulated transients and failure domain identification and characterization in SARNET benchmark exercise [8].

In this paper we extend our previous work [9] by improving the methods, providing detailed description of the approaches. Specifically, the clustering algorithms and visualization techniques for decision trees have been significantly improved with respect to [9]. In addition, we consider application of developed methods in decision support context.

In Section 2 we provide general description of the approach. In Section 3 we describe a hypothetical accident scenario in a typical French design of Pressurized Water Reactor (PWR). An example of application of the proposed approach to the selected accident scenario is presented in Section 4, followed by the discussion and conclusions.

2. Classification Approach

Methodologies that take into account uncertainty in timing of events can produce potentially unlimited number of transient scenarios for a single initiating event. For decision making, handling of the huge amount of data is a challenge. The development of insights and understanding requires interpretation of the scenario evolutions in order to identify the principal characteristics of the events that contribute to the risk. In order to solve this problem we develop an approach based on clustering and decision trees for explaining the structure of the clustered data (see Figure 1).

The main steps of this approach are briefly explained below. Firstly, the scenario grouping is performed (see Section 2.1). The main idea of this step is to focus the analysis on the sequences intractable in classical PSA. Thus, scenarios where the order and timing of events are not important are grouped first and excluded from further considerations as those directly amenable to PSA analysis. Then we group scenarios where the order of events is important but not their timing. Remaining group of scenarios contains sequences where the outcome depends on the order and timing of the events.

Next, Principal Component Analysis (PCA) [10, 11] is carried out in order identify and quantifying a group of principal components which have the largest influence on the system response (see Section 2.2). Then, based on the PCA results the clustering analysis is performed using Adaptive Mesh Refinement (AMR) method (see Section 2.3.1). In the final step a decision tree is built for each failure mode using clustering results data [12]. Decision tree is used for data representation that explains failure domain-cluster structure (see Section 2.4). The structure is easy to visualize and interpret in the decision-making process. Finally, information of the leaf nodes is used for failure domain probability calculation. Decision tree classification algorithm performs orthogonal partitioning of the search space using data impurity measure as a splitting criterion [10, 13].
2.1. Scenario Grouping. System codes are used in IDPSA in order to evaluate temporal evolution of the accident progression for different time dependent sequences of the events such as activation or failure of safety systems (e.g., reactor protection system and emergency core cooling system). The main purpose of scenario grouping is to identify and separate sequences of events that can be treated in classical PSA, that is, those where order and timing of events have no effect on the outcome (safe or failure end state). The approach is represented in Figure 2.

The numeric algorithm used in scenario grouping is similar to those used in sequence pattern analysis [14]. Each event is represented by a unique number. Thus each simulated transient is represented by a sequence of numbers. Then, for the whole data set, all possible patterns are identified and split into two categories with the same (1) sets of events and (2) order of events. It is important to note that the first category can contain several patterns of the second category (e.g., the set [2, 3] in the first category will represent sequences (2, 3) and (3, 2) in the second category). Then the following steps of the grouping algorithm are performed:

1. The sets of events that always lead to either failure or safe condition are identified for further treatment in PSA. If the same set of events can lead to both failure and safe states it means that timing and/or order of events can be important. Such sets of events are treated further in Steps (2) and (3).

2. The sequences of events which always lead to either failure or safe condition are identified. If the same sequence of the events can lead to both failure and safe conditions it is a sign that the influence of timing of the events is important.

3. The sequences of events where outcome depends on the timing of the events and parameter uncertainty and requires respective dynamic treatment are considered further in the following steps of the analysis, that is, PCA and data transformation, Scenario Clustering, and so forth (see also Figure 1).

2.2. Principal Component Analysis. Principal Component Analysis (PCA) is a technique for revealing the relationships between variables in a data set by identifying and quantifying a group of principal components. These principal components are composed of transformations of specific combinations of input variables that relate to a given output (or target) variable [11]. Each principal component accounts for a decreasing amount of the variations in the raw data set; that is, the first principal component is responsible for the largest possible variance (accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it has to be orthogonal to (i.e., uncorrelated with) the preceding components.

The main purpose of application of PCA in the classification approach is to transform the data without rescaling into a new orthogonal coordinate system that optimally describes the variance in a single dataset. The data transformation is defined by

\[ X^{*T} = X^{T} W, \]

where \( X^{*T} \) and \( X^{T} \) are the new and old vectors of observations and \( W \) is the matrix of principal component coefficients (eigenvectors of the covariance matrix \( XX^{T} \)) [11].

2.3. Scenario Clustering. The purpose of clustering analysis is to assign members to each group such that members of a group are more similar (according to specific criteria) to each other than to those in other groups (clusters). Clustering analysis is the task of grouping a set of objects in a way that objects within one group (or cluster) are more similar than those in the other groups. It can be achieved by various algorithms that can differ significantly in their notion of what constitutes a cluster and how to efficiently find them. There are several clustering algorithms that methodologically can be separated into connectivity models (hierarchical clustering [15]), centroid based clustering (K-means [15]), distribution based clustering, density based clustering [16], artificial neural networks [17], fuzzy clustering, clustering methodologies based on evolutionary algorithms (Genetic Algorithms [18]), and grid based clustering methodologies [12]. The methodology presented in this paper is based on grid based clustering algorithms with adaptive mesh refinement [12, 19].

2.3.1. Grid Based Clustering. Grid-based clustering methods partition the space into a finite number of cells that form a grid structure on which all of the operations for clustering are carried out. The main advantage of the approach is its computational efficiency [19–21].

Given a set of \( n \)-dimensional data and the input parameter, cell size, the search space is partitioned into nonoverlapping rectangular \( n \)-dimensional units (cells) of the size \( \xi \). For the sake of conservatism we do not use density threshold for the unit's selectivity parameter (amount of scenarios contained in the unit). Although it might be used in the future development with adaptation of adaptive mesh refinement (AMR) algorithms under conservatism constraints no failure scenarios can be identified as an outlier [19].

Once grid is defined, the algorithm looks for the clusters of cells that contain failure scenarios of the same failure mode. Two cells can form a cluster if they have a common face. The algorithm presents large amount of scenarios with different failure modes as a finite number of cells grouped into clusters corresponding to the same failure mode.

Mesh Refinement. In the adaptive mesh refinement technique the algorithm starts with initial coarse grid. Then, the algorithm identifies the regions with transition between "safe"
and “failure” and introduces higher resolution subgrids only in those regions. Finer subgrids are added recursively until either a given maximum level of refinement is reached or the local resolution criterion for the boundary between “safe” and “failure” regions is achieved. Thus in an adaptive mesh refinement computation grid spacing is fixed for the base grid only and is determined locally for the subgrids according to the requirements of the problem.

2.4. Application of Decision Trees. A grid based clustering algorithm performs orthogonal partitioning of the uncertainty space, similar to the partitioning of learning data set in the decision tree. Therefore, complexity of the decision trees can significantly reduce when using clustering results data rather than raw scenario data.

A decision tree is a classification and data-mining tool for extraction of useful information contained in large data sets. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated recursively for the subtree rooted at the new nodes until no further branching in the tree can be made or some stopping preset conditions are met [10, 13]. A flow-chart-like structure is generated in which internal nodes represent test on an attribute, each branch represents outcome of test and each leaf node represents class label (decision taken after computing all attributes). Decision trees can be used as a powerful visual and analytical decision support tool; especially in case of multidimensional data, visualization of results in the original space is nontrivial. Decision tree can be constructed using different data impurity measures (e.g., Gini impurity measure and information gain measure) to select the best split among the candidate attributes at each step while growing the tree [13]. Decision trees also can be used as a predictive model which maps observations about an item to conclusions about the item’s target value.

2.4.1. Classification and Regression Decision Trees. Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees [10, 22]. The best split is identified by a splitting criterion that uses different data impurity measures (e.g., Gini impurity and information gain measure). In this work we use Classification and Regression Tree (CART) with Gini criterion. CART is a nonparametric decision tree learning technique that produces either classification or regression trees, depending on whether the dependent variable is categorical or numeric, respectively [23].

The Gini impurity index (commonly used in CART) at node $t$ is defined as

$$Gini(t) = \sum_{j \neq i} p(j | t) \cdot p(i | t)$$  \hspace{1cm} (2)

where $i$ and $j$ are the categories of the target variable, $p(j, t)$ and $p(i, t)$ are proportion of cases in node $t$ with attributes $i$ and $j$, respectively. Thus, when the cases in a node are evenly distributed across the target categories, the Gini index takes its maximum value $1 - 1/k$, where $k$ is the number of categories for the target variable. The minimum value is zero and it occurs when all the data at a node belongs to one target category.

The Gini criterion for split at $s$ at a node $t$ is defined as

$$Gini_{split}(s, t) = Gini(t) - p_L \cdot Gini(t_L) - p_R \cdot Gini(t_R),$$  \hspace{1cm} (3)

where $p_L$ is the proportion of cases in $t$ sent to the left child node and $p_R$ is the proportion of cases in $t$ sent to the right child node. $s \in S$ refers to a particular generic split among all possible sets of splits $S$. 

**Figure 2: Scenario grouping algorithm.**
The split $s$ is chosen to maximize the value of $\text{Gini}_{\text{split}}(s, t)$. Since $\text{Gini}(t)$ is constant for any split $s$ on node $t$, it can be alternately said that the split $s$ is to be chosen such that the quantity

$$\text{Gain}(s, t) = p_L \text{Gini} (t_L) + p_R \text{Gini} (t_R) \quad (4)$$

is minimized [23].

2.4.2. Probability Estimation Using Decision Trees. The failure domain is represented by agglomerations (clusters) of nonoverlapping cells (grids) in the uncertainty space. If all points in the uncertainty space are equally probable then the probability of the failure domain is the ratio of the volume of the failure domain to the total volume of the uncertainty space.

Decision tree represents the failure domain by final nodes in the tree and respective classification rules that lead to these nodes. The probability of each cell can be obtained as average probability of scenarios contained in correspondent cell:

$$\overline{P}_k = \frac{\sum_{i=1}^{N} p_i}{N_{\text{scene}}} \quad (5)$$

and the probability of a failure mode $i$ is

$$p_i = \sum_{j=1}^{N} \sum_{k=1}^{M_j} \overline{P}_k \xi^n_j,$$  

where $n$ is dimensionality, $\xi^n_j$ is cell volume, $\overline{P}_k$ is average probability of scenarios contained in cell $k$, $M_j$ are cells contained in the final failure node (leaf) $j$, and $N$ is total amount of failure nodes (leafs). Depending on the values $\overline{P}_k$ it is possible to assign weights per each cell when building a tree, so the scenarios (cells) with higher probability are likely to be classified into the same final node.

3. Application

In order to illustrate proposed approach we chose a benchmark exercise developed in the framework of the SARNET exercise see [8]. For determining the limit of inflammability for the gas mixture Shapiro diagram is used (see Figure 3).

Table I gives the limit for inflammability in terms of molar fractions of $\text{H}_2$ versus $\text{H}_2\text{O}$.

Water Injection. If water injection occurs before total core uncovery (5875 s), it is assumed that little hydrogen is produced and the vessel rupture is avoided. The probability of this scenario is 0.5.

The probability that water injection is available between total core uncovery (5875 s) and vessel rupture (14220 s) is 0.5. The probability of water injection initiation timing is uniformly distributed in the time interval between total core uncovery and vessel rupture.

Spray System Activation. The probability that the spray system can be activated after core uncovery (4080 s) and before vessel rupture is equal to 0.5. If the spray system can be activated, the probability of spray system activation is uniformly distributed in the time interval between core uncovery (4080 s) and vessel rupture.

Delay before Combustion. A delay before combustion becomes shorter as $\text{H}_2$ concentration increases. To determine this delay, the following rules are used [8]:

(i) If hydrogen concentration ($\text{H}_2 < \text{hydrogen inflammability limit (H}_{21\text{IF}}$), no combustion can occur.
(ii) If $\text{H}_2 = \text{H}_{21\text{IF}}$ inflammability limit, the probability of delay before the first (or after previous) combustion is uniformly distributed between 0 and 4 hours.
(iii) If $\text{H}_2 \geq \text{hydrogen ignition limit (H}_{2\text{IG}}$, the probability of delay before the first (or after previous) combustion is uniformly distributed between 0 and 20 minutes.
Table 1: Limit for inflammability.

<table>
<thead>
<tr>
<th>Molar fraction of H$_2$O, %</th>
<th>Inflammability limit for H$_2$ molar fraction, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>4.5</td>
</tr>
<tr>
<td>20</td>
<td>5.5</td>
</tr>
<tr>
<td>30</td>
<td>6.7</td>
</tr>
<tr>
<td>40</td>
<td>8.1</td>
</tr>
<tr>
<td>50</td>
<td>10.1</td>
</tr>
</tbody>
</table>

If $H_{2IF} < H_2 < H_{2IG}$, the probability of delay before first (or after previous) combustion is uniformly distributed between 0 and $\Delta T_{max}$ (see (7)):

$$\Delta T_{max} (H_2) = \frac{4(H_2 - H_{2IG}) - 0.333 (H_2 - H_{2IF})}{H_{2IF} - H_{2IG}}.$$  (7)

In this work we consider only containment pressure of $P_{Lim} = 0.3$ MPa threshold as a failure criterion for the sake of simplicity. Using Monte Carlo sampling over 443200 scenarios has been generated for INI (initiating event) + all possible combinations of SIS, CHRS, and IGNI, with different timing of these events.

4. Results

Performing grouping analysis we identified the following possible sequences of the events: [INI SIS]; [INI SIS CHRS]; [INI SIS IGNI]; [INI SIS CHRS IGNI]; [INI CHRS SIS]; [INI CHRS IGNI]; [INI CHRS SIS IGNI]; [INI CHRS IGNI SIS]. Classification analysis suggests that sets of events [INI, CHRS], [INI, SIS], [INI, CHRS, SIS], and [INI, SIS, CHRS] do not cause containment over pressurization when they are not followed by hydrogen ignition event (IGNI). Sequences [INI CHRS IGNI] and [INI CHRS IGNI SIS] also do not generate pressure spike big enough to cause containment failure. In the sequences [INI SIS IGNI], [INI SIS CHRS IGNI], and [INI CHRS SIS IGNI] the outcome depends on the timing of ignition (IGNI) and safety systems actuation (see Table 2 for conditional containment failure probabilities for these sequences). In Figure 4 we illustrate an example of application of clustering analysis and decision trees for the sequences that require dynamic treatment.

The advantage of using PCA and coordinate system defined by the principal components of the failure domain is that it significantly reduces the complexity of the decision tree. In case of the transformed coordinate system the decision tree was able to characterize almost 50% of the data set separating the major part of failure scenarios from safe scenarios only in 2 cuts. The results can be transferred back into original coordinate system simply by inverting (1) as follows:

$$X^T = (X^*T - C)W^T,$$  (8)

where $W$ is orthogonal matrix ($W^T = W^{-1}$) with principal component coefficients (eigenvectors of the covariance matrix $XX^T$). In this particular case the values of the $W$
matrix correspond to ~18.2 degrees rotation counterclockwise, and the variables are defined through the linear combination of variables in original coordinate system:

\[
SIS^* = 0.95 \times SIS + 0.31 \times IGNI + 0.31,
\]

\[
IGNI^* = -0.31 \times SIS + 0.95 \times IGNI - 0.007.
\]  

The new variables represent linear combinations of all the original parameters involved. The decision tree rules (e.g. SIS* > 2955 sec) in new variables can be also interpreted in the original coordinate system.

Figures 5 and 6 illustrate the results of clustering analysis for the sequence [INI SIS IGNI] with uniform grid. The cells that contain failure scenarios are grouped into cluster representing the failure domain. For each cell in the cluster the algorithm calculates correspondent probability of failure (Figure 7).

Different values of probabilities in the different parts of the failure domain correspond to different H₂ concentrations and respective probability distributions for the time delays of ignition event [8]. For instance, in Figure 8, H₂ concentration is below ignition limit and above inflammability limit; therefore the time delay before the first combustion is uniformly distributed between 0 and \(\Delta T_{\text{max}}(H_2)\) (see (7)). In Figure 9, H₂ concentration is above its inflammability and
ignition limits, therefore, according to [8], time delay before combustion is uniformly distributed between 0 and 20 mins.

Failure domain structure can be represented using clustering data and decision tree. To illustrate the approach and to provide a possibility to compare failure domains, presented in Figures 5 and 6, the results are visualized with the decision trees. In this work we use limited amount of uncertain parameters for the sake of visual comparison of the data representation; however, the main advantage of the decision tree approach is the ability to represent complex failure domains with four or more uncertain parameters, when it is difficult to visualize results using other methods. Decision tree complexity depends on the shape of the failure domain and level of details (initial grid and refinement step). However, it is possible to prune decision tree, so the complexity and precision are kept in acceptable levels. Pruning is the process of reducing a tree by turning some branch nodes into leaf nodes and removing the leaf nodes under the original branch [24]. Trees are pruned based on an optimal pruning scheme that first prunes branches giving less improvement in error cost.

After computing an exhaustive tree, the algorithm eliminates nodes that do not contribute to the overall prediction, decided by another essential ingredient, the cost of complexity. This measure is similar to other cost statistics, such as

Figure 5: Cluster representation of the failure domain (red) and safety domain (green) for the sequence [INI SIS IIGNI], axes scaled between 0 and 1.

Figure 6: Cluster representation of the failure domain (red) and safety domain (green) for the sequence [INI SIS IIGNI] (SIS, IIGNI, in coordinate system defined by principal components of the dataset), axes scaled between 0 and 1.

Figure 7: Containment failure probability distribution for sequence [INI SIS IIGNI].

Figure 8: H₂ molar fraction (%) and H₂ inflammability and ignition limits (%).
Figure 9: H₂ molar fraction (%) and H₂ inflammability and ignition limits (%).

Figure 10: Decision tree fitted into clustering results data for the sequence [INI SIS IGNI] (sec) with pruning.
Mallows’ $C_p$ [25], which adds a penalty for increasing the number of parameters in a model [24].

Decision tree results for the sequence [INI SIS IGNI] indicate that containment failure is possible if IGNI$^*$ event occurs in the time window between 1230.55 and 4444.07 sec (in coordinate system defined by principal components of the dataset). Depending on the timing of the occurrence of the events, H$_2$ combustion within this time window can challenge containment integrity.

The pruning (cutting) in the decision trees is done at the point where the further refinement will not improve the results and, on the other hand, increase the complexity of the
**Figure 14:** Decision tree fitted into clustering results data for the sequence [INI SIS CHRS IGNI] (sec) for controllable variables.

decision tree. Decision trees (Figures 10 and 11) are built with

data set in both original coordinate system and coordinate system
defined by its principal components (Figures 5 and 6).

4.1. Decision Support Model. Let us consider as an example

of sequence [INI SIS CHRS IGNI]. Figure 12 shows cluster

representation of the failure domain in this sequence.

When it comes to decision support, H\textsubscript{2} ignition event

(IGNI) in this sequence is entirely stochastic event; that is,
operator has no control over it. On contrary, water

injection (SIS) and containment spray (CHRS) systems can
be actuated by operator at specified moment of time and,
therefore, they are controllable. Decision trees can be used

to build decision support model based on the controllable

events; that is, decision trees can help us to find an answer to

the question "what can be done in case of LOCA initiating

event to avoid containment failure?". Figure 13 illustrates

failure domain for the sequence [INI SIS CHRS IGNI] in
terms of controllable events SIS and CHRS. Based on
the clustering results we build a decision tree in variables
representing time delays for actuation of the safety systems
(SIS and CHRS) and correspondent outcome (Figures 14
and 15). Obtained results indicate that for the sequence

[INI SIS CHRS IGNI] containment failure can be avoided
in case of early actuation of water injection and containment
spray systems (in the range of ∼492 seconds) or in case of
late activation of containment spray (over ∼4000–6944 sec
depending on the actuation time of water injection).

5. Discussion

In this work we present an approach for grouping and classi-

fication of typical "failure/safe" scenarios identified using

IDPSA methods. This approach allows the classification of
scenarios that are directly amenable in classical PSA and
scenarios where order of events, timing, and parameter

uncertainty affect the system evolution and determine viola-
tion of safety criteria.

We use grid based clustering with AMR and decision
trees for characterization of the failure domain. Clustering
analysis is used to represent the failure domain as a finite
set of the representative scenarios. Decision trees are used to
visualize the structure of the failure domain. Decision trees
can be applied to the cases where four or more uncertain
parameters are included in the analysis and it is difficult to
visualize results in three-dimensional space.

Proposed approach helps to present results of the IDPSA
analysis in a transparent and comprehensible form, amenable
to consideration in the decision-making process. Useful
insights into the complex accident progression logic can be
obtained and used for development of understanding and
mitigation strategies of the plant accidents including severe
accidents. The insights can be employed to reduce unnec-

essary conservatism and to point out areas with insufficient
conservatism in deterministic analysis. Results of the analysis
can be also used to facilitate connection between classical
PSA and IDPSA analysis.
Conflicts of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study was supported by the Swedish Radiation Safety Authority (SSM). The authors are grateful to Dr. Wiktor Frid (SSM) for very useful discussions.

References

Research Article

An Enhanced Preventive Maintenance Optimization Model Based on a Three-Stage Failure Process

Ruifeng Yang, 1 Jianshe Kang, 1 and Zhenya Quan 2

1 Mechanical Engineering College, No. 97, Heping West Road, Shijiazhuang, Hebei 050000, China
2 Shanxi Conservancy Technical College, Taiyuan, Shanxi 030000, China

Correspondence should be addressed to Ruifeng Yang; rfyangphm@163.com

Received 14 January 2015; Revised 1 March 2015; Accepted 16 March 2015

Academic Editor: Francesco Di Maio

Copyright © 2015 Ruifeng Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nuclear power plants are highly complex systems and the issues related to their safety are of primary importance. Probabilistic safety assessment is regarded as the most widespread methodology for studying the safety of nuclear power plants. As maintenance is one of the most important factors for affecting the reliability and safety, an enhanced preventive maintenance optimization model based on a three-stage failure process is proposed. Preventive maintenance is still a dominant maintenance policy due to its easy implementation. In order to correspond to the three-color scheme commonly used in practice, the lifetime of system before failure is divided into three stages, namely, normal, minor defective, and severe defective stages. When the minor defective stage is identified, two measures are considered for comparison: one is that halving the inspection interval only when the minor defective stage is identified at the first time; the other one is that if only identifying the minor defective stage, the subsequent inspection interval is halved. Maintenance is implemented immediately once the severe defective stage is identified. Minimizing the expected cost per unit time is our objective function to optimize the inspection interval. Finally, a numerical example is presented to illustrate the effectiveness of the proposed models.

1. Introduction

It is well recognized that nuclear power plants (NPPs) are tightly regulated complex systems, where the issues related to their safety are of primary importance [1]. A possible and widely used way of analyzing the performance of safety for NPPs is to perform probabilistic safety assessments (PSA). PSA constitutes the most widespread methodology for studying the safety of NPPs and plays an increasing role in safety analysis and in developing safety improvements for industrial systems [2]. One of the important tasks of PSA for NPPs is evaluating the system reliability in terms of quantifiable measures. It is well known that there are many factors that affect the total reliability of NPPs. Among them, maintenance has been considered as one of the most important factors [3, 4].

The purpose of maintenance in nuclear applications is to identify and mitigate the degradation of plant or system, as well as to restore the functions of system to an acceptable level, thus improving the reliability of NPPs. The optimization of maintenance programs could be summarized as “the right work, on the right equipment, at the right time” [5]. Here the “right time” refers to scientific and appropriate decision-making for initiating a maintenance program [6].

In this paper, our aim is to develop maintenance models to optimize the interval of maintenance programs in terms of “right time.” Numerous maintenance policies have been implemented on the NPPs to prevent the occurrence of failure [7]. Preventive maintenance (PM) is perhaps one of the most popular maintenance policies, by which maintenance activities are executed with a planned interval aiming at preventing potential failures from occurring [8, 9]. PM is still a dominant maintenance policy due to its easy implementation in industrial applications.

As we all know, inspection as an important PM activity could reveal the status of the system being inspected; thus it helps maintenance engineers make decisions to avoid the occurrence of failure [10]. Inspections may be performed discretely with a periodic or aperiodic interval by using inspection instrument or continuously by modern condition
monitoring devices. The periodic inspection activities are common in industry because of the effectiveness and convenience.

However, how often do engineers inspect the plants or the determination of inspection interval is still a key issue. The traditional determination of inspection interval is decided by managers’ experience. Of course it is not scientific. Many researchers have developed numerous PM optimization models to search the optimal inspection interval under various modeling conditions [11–13]. But the models based on the delay time concept have been proved to have the obvious advantages for optimizing the inspection interval since the delay-time technique can directly model the relationship between the inspection intervention and the system performance in contrast with other PM models [14, 15].

The delay time concept was first introduced by Christer in 1976 [16], which defines the failure process as a two-stage failure process, namely, normal stage and delay time stage. The normal stage is from new to an initial point that a defect can be first identified by an inspection and the delay time stage from this initial point to failure. By the definition of delay time concept, the plant can be in one of two states before failure, namely, normal and defective. A defect can be identified if the inspection is carried out during the delay time stage. Many inspection and PM models, especially successful case studies based on the two-stage failure process, have been reported with actual applications in industry [17].

However, in industrial applications, the systems may be described more than two states before failure as the three-color scheme is mostly used to divide the state before failure into green (normal), yellow (need attention), and red (need immediate attention) [18]. Considering this industrial scenario, Wang [19] firstly extended the two-stage failure process into a three-stage failure process, which is closer to the practical situation and provides more decision options for different states. In the work [19], the inspection interval is shortened to be half of the current interval to more frequently inspect the system when the minor defective stage is identified but immediately replace the system if it is in the severe defective stage. Wang et al. [20] further considered a two-level inspection policy with PM and delaying the maintenance once the severe defective stage is identified and the time interval to the next PM is less than a threshold level. Yang et al. [21, 22] proposed an inspection maintenance optimization model with imperfect maintenance based on a three-stage failure process. An inspection optimization model based on a three-stage failure process is presented by Yang et al. [23].

The contribution that Wang extended the failure process into three stages is meaningful, as this extension is closer to reality and corresponds to the industrial scenario. When the minor defective stage is identified, halving the inspection interval is adopted. However, one may argue that how often do engineers halve the inspection interval. It is done only when the minor defective stage is identified at the first time or if only identifying the minor defective stage because the former may be easily implemented for maintenance operators but the latter can inspect the condition of system frequently and avoid the failure to some extent. This question is not explained in the previous work based on the three-stage failure process.

Therefore, in this paper we propose an enhanced preventive maintenance optimization model based on a three-stage failure process to improve the reliability of systems, thus laying the foundation for safety assessment of NPPs. The lifetime of system before failure is divided into three stages, namely, normal, minor defective, and severe defective stages. When system fails, replacement is carried out. Once the severe defective stage is identified, the maintenance is implemented immediately. But when the minor defective stage is identified, we adopt two measures for comparison: one is that halving the inspection interval only when the minor defective stage is identified at the first time; the other one is that if only identifying the minor defective stage, the subsequent inspection interval is halved. Minimizing the expected cost per unit time is our objective function to optimize the inspection interval.

The contributions of this paper are as follows: (1) an enhanced preventive maintenance optimization model based on the three-stage failure process is presented to lay the foundation for safety assessment of NPPs; (2) the deterioration of system is subject to the three-stage failure process; (3) two different maintenance decisions are considered for comparison when the minor defective stage is identified.

The remaining parts of the paper are organized as follows: Section 2 presents the modeling assumptions and notation. Section 3 provides the proposed cost models with perfect inspection case and imperfect inspection case is modeled in Section 4. Section 5 gives numerical examples. Finally, Section 6 concludes the paper and the future researches are suggested.

2. Modeling Assumptions and Notation

The following assumptions and notation are presented for the subsequent modeling.

1. The failure process of single component system is divided into three stages: normal stage $U$, minor defective stage $V$, and severe defective stage $H$. These three stages are assumed to be independent.

2. The system is inspected periodically with an interval $t$.

3. Both perfect and imperfect inspection cases are considered. Perfect inspection can always reveal the defective stage no matter it is minor and severe. However, imperfect inspection may miss the minor defective stage with a probability of $(1 - r)$ but always can identify the severe defective stage.

4. If the system is found in the normal stage, do nothing. Failure can be observed immediately and replacement is always carried out at once.

5. Once system is found to be in the severe defective stage, it is always repaired immediately.

6. As to the minor defective stage, two actions are considered here: the first one is to shorten the inspection interval only when the minor defective stage is
identified at the first time; the other one is to shorten the inspection interval if only the minor defective stage is identified.

(7) Repair or replacement is regarded as renewing the system, though it may be the only option for a single component system.

The subsequent modeling will use the following notations:

\( U, f_U(u), \) and \( F_U(u) \): random variable representing the duration of normal stage, probability density function (pdf), and cumulative distribution function (cdf) of normal stage;

\( V, f_V(v), \) and \( F_V(v) \): random variable representing the duration of minor defective stage, pdf, and cdf of minor defective stage;

\( H, f_H(h), \) and \( F_H(h) \): random variable representing the duration of severe defective stage, pdf, and cdf of severe defective stage;

\( t \): inspection interval;

\( T_f \): random failure time;

\( T_p \): random time of an inspection renewal due to the severe defective stage identification by an inspection;

\( r \): identification probability of the minor defective stage with imperfect inspection;

\( C_s \): average cost per inspection;

\( C_f \): average cost per failure;

\( C_p \): average cost caused by an inspection renewal due to identification of the severe defective stage;

\( EC(t) \): expected renewal cycle cost;

\( EC_f(t) \): expected cost caused by failure renewal;

\( EC_p(t) \): expected cost caused by inspection renewal due to identification of severe defective stage;

\( EL(t) \): expected renewal cycle length;

\( EL_f(t) \): expected length caused by failure renewal;

\( EL_p(t) \): expected length caused by inspection renewal due to identification of severe defective stage;

\( C(t) \): expected cost per unit time.

3. Cost Models with Perfect Inspection

Two cost models are formulated in this section based on the different decisions when identifying the minor defective stage. Perfect inspection means that inspections can identify the defective stages no matter the defect is minor or severe. The objective function of the models is to minimize the expected cost per unit time for deriving the optimal inspection interval. There are two renewal scenarios, namely, a failure renewal and an inspection renewal caused by identifying the severe defective stage. According to the renewal theory, the renewal probabilities need to be derived before formulating these cost models.

Model 1. The inspection interval is shortened only when the minor defective stage is identified at the first time.

3.1. The System May Be Renewed before the Minor Defective Stage Is Identified. There could exist two renewal scenarios, as shown in Figure 1.

(1) A failure renewal happens when the system deteriorates into the minor defective stage and then fails in \(((i-1)t, it)\) simultaneously. As no minor defective stage is identified, the total length of minor defective and severe defective stage is shorter than the inspection interval \(t\). The probability of such a failure renewal is

\[
P((i-1)t < T_f < it) = P((i-1)t < U < it, 0 < V < it - U, 0 < H < it - U - V) \]

\[
= \int_{(i-1)t}^{it} f_U(u) \int_{0}^{it-u} f_V(v) F_H(it - u - v) \, dv \, du.
\]

(2) An inspection renewal occurs once the severe defective stage is found at \(it\) by an inspection. The minor defective stage is within \(((i-1)t, it)\) and severe defective stages are longer than \(it - u - v\), where \(u\) and \(v\) are the duration of the normal and minor defective stages. The probability of such an inspection renewal is

\[
P(T_p = it) = P((i-1)t < U < it, 0 < V < it - U, H > it - U - V)
\]

\[
= \int_{(i-1)t}^{it} f_U(u) \int_{0}^{it-u} f_V(v) (1 - F_H(it - u - v)) \, dv \, du.
\]

According to renewal probabilities, the expected renewal cycle cost and length with different renewal scenarios can be derived.
### 3.1.1. Expected Renewal Cycle Cost $EC_1(t)$

If the system fails in $((i-1)t, it)$ and failure renewal happens, the relevant cost is $(C_f + (i-1)C_s)$, including the cost of $(i-1)$th inspection and failure replacement cost. But once the severe defective stage is identified at $it$ and inspection renewal happens, the relevant cost is $(C_p + iC_s)$, including the cost of $ith$ inspection and maintenance cost. The relevant cost of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle cost $EC_1(t)$ can be derived as

$$EC_1(t) = \sum_{i=1}^{\infty} \left( (C_f + (i-1)C_s) P((i-1)t \leq T_f < it) + (C_p + iC_s) P(T_p = it) \right)$$

$$= \sum_{i=1}^{\infty} \left( (C_f + (i-1)C_s) \int_{(i-1)t}^{it} f_U(u) \int_0^{it-u} f_V(v) f_H(it-u-v) dvdu 
+ (C_p + iC_s) \int_{(i-1)t}^{it} f_U(u) \int_0^{it-u} f_V(v) (1 - F_H(it-u-v)) dvdu \right).$$

### 3.1.2. Expected Renewal Cycle Length $EL_1(t)$

Before deriving the expected cycle length of failure renewal, the pdf of system fails at $T_f$, $T_f < P((i-1)t + h < T_f < (i-1)t + h + dh)$ should be formulated firstly using the same way as (1)

$$P((i-1)t + h < T_f < (i-1)t + h + dh)$$

$$= \int_{(i-1)t}^{it} f_U(u) \int_{it-h}^{it} f_V(v) f_H((i-1)t + h - u - v) dvdu.$$

The expected renewal cycle length can be derived as the way as (3). The relevant length of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle length $EL_1(t)$ can be derived as

$$EL_1(t) = \sum_{i=1}^{\infty} \left( \int_0^{(i-1)t} ((i-1)t + h) \right)$$

$$= \sum_{i=1}^{\infty} \left( \int_0^{(i-1)t} ((i-1)t + h) \right)$$

### 3.2. The System Is Renewed after the Minor Defective Stage Is Identified at $it$

There also exist two renewal scenarios, as shown in Figure 2.

1. The system fails in $(it + (n-1)t/2, it + nt/2)$ after the minor defective stage is identified at $it$. The minor defective stage starts within $((i-1)t, it)$ and ends within $(it + (n-1)t/2, it + nt/2)$. The severe defective stage and failure happens in the same interval. The probability of such a renewal is

$$P((i-1)t < U < it, it + \frac{(n-1)t}{2} < T_f < it + \frac{nt}{2})$$

$$= P(it + \frac{(n-1)t}{2} < U < it, it + \frac{(n-1)t}{2} < U < V < it + \frac{nt}{2} < U, 0 < H < it + \frac{nt}{2} - U - V)$$

2. The system is renewed at $it + nt/2$ as the severe defective stage is firstly identified. The normal stage ends within $((i-1)t, it)$. The minor defective stage is firstly identified at $it$ and ends within $(it + (n-1)t/2, it + nt/2)$. So the severe defective stage is longer than $it + nt/2 - U - V$. The probability of such a renewal is

$$P\left(T_p = it + \frac{nt}{2}\right)$$

$$= P\left((i-1)t < U < it, it + \frac{(n-1)t}{2} < U < V < it + \frac{nt}{2} - U, H > it + \frac{nt}{2} - U - V\right).$$
Failure renewal
Inspection renewal

Figure 2: The system is renewed after the minor defective stage is identified at the first time.

\[
\int_{(i-1)t}^{it} f_U(u) \int_{it+(n-1)t/2}^{it+nt/2-u} f_V(v) \cdot \left(1 - F_H \left( \frac{it + nt}{2} - u - v \right) \right) dv du.
\]

Therefore, the expected renewal cycle cost and length under different renewal scenarios can be given as follows.

### 3.2.1. Expected Renewal Cycle Cost \( E_C(t) \)

When the system fails in \((it + (n-1)t/2, it + nt/2)\), the relevant cost is \((C_f + (i + n - 1)C_s)\), including the cost of \((i + n - 1)\)th inspection and failure replacement cost. But once the severe defective stage is identified at \(it + nt/2\), the relevant cost is \((C_p + (i + n)C_s)\), including the cost of \((i + n)\)th inspection and maintenance cost. The relevant cost of each renewal event multiplies by the renewal probability, respectively; thus the expected renewal cycle cost \( E_C(t) \) can be derived as

\[
E_C(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( C_f + (i + n - 1)C_s \right) 
\cdot \int_{(i-1)t}^{it} f_U(u) \int_{it+(n-1)t/2}^{it+nt/2-u} f_V(v) \cdot \left(1 - F_H \left( \frac{it + nt}{2} - u - v \right) \right) dv du + (C_p + (i + n)C_s) \cdot \int_{(i-1)t}^{it} f_U(u). \]

### 3.2.2. Expected Renewal Cycle Length \( EL(t) \)

The relevant length of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle length \( EL(t) \) can be derived as

\[
EL(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( \int_{0}^{(i+1/2)t} \left( it + \frac{(n-1)t}{2} + h \right) \int_{(i-1)t}^{it} f_U(u) \int_{it+(n-1)t/2}^{it+nt/2-u} f_V(v) f_H \cdot \left( \frac{it + nt}{2} + h - u - v \right) dv du dh + \left( \frac{it + nt}{2} \right) \int_{(i-1)t}^{it} f_U(u) \int_{it+(n-1)t/2}^{it+nt/2-u} f_V(v) \cdot \left(1 - F_H \left( \frac{nt}{2} - u - v \right) \right) dv du. \right)
\]

### 3.2.3. Expected Cost per Unit Time \( C(t) \)

Based on the expected cycle cost and length, expected cost per unit time can be calculated using the renewed reward theory [24] as

\[
C(t) = \frac{E_C(t)}{EL(t)} = \frac{E_C(t) + E_C(t)}{EL(t) + EL(t)}. \]

Model 2. The inspection interval is shortened if only the minor defective stage is identified.

3.3. The System Is Renewed before Founding the Minor Defective Stage. This scenario is the same as Section 3.1.

3.4. The System Is Renewed after Identifying the Minor Defective Stage. As this model supposes that if only the minor defective stage is identified, the subsequent inspection interval is halved. Here, we consider the scenario that system is renewed after the minor defective stage is identified at \(k\)th time and the inspection interval is shortened to be \(t/2^k\) for generality.
The system deteriorates into the minor defective stage within \((i-1)t, it)\) and the minor defect is firstly identified at \(it\) by an inspection. So the subsequent inspection interval is halved. Finally, the minor defective stage ends within \((t_k, t_k^*)\) and the severe defective stage and failure happens in the same interval. The probability of such a failure renewal is

\[
P(t_k < T_f < t_k^*)
\]

\[
= P((i-1)t < U < it, \ t_k - U < V < t_k^* - U,
\]

\[
0 < H < t_k^* - U - V)
\]

\[
= \int_{(i-1)t}^{it} f_U(u) \int_{t_k-u}^{t_k^*-u} f_V(v) f_H(t_k^* - u - v) \, dv \, du.
\]

(11)

(2) The system is replaced when the inspection renewal happened. The minor defective stage is firstly identified at \(it\) and ends within \((t_k, t_k^*)\). The severe defective stage is longer than \(t_k^* - U - V\) and identified at \(t_k^*\), leading to the system renewal. The probability of this inspection renewal is

\[
P(T_p = t_k^*)
\]

\[
= P((i-1)t < U < it, \ t_k - U < V < t_k^* - U,
\]

\[
H > t_k^* - U - V)
\]

\[
= \int_{(i-1)t}^{it} f_U(u) \int_{t_k-u}^{t_k^*-u} f_V(v) (1 - F_H(t_k^* - u - v)) \, dv \, du.
\]

(12)

Accordingly, the expected renewal cycle cost and length under different renewal scenarios can be given using the same way as model 1.

### 3.4.1. Expected Renewal Cycle Cost \(EC_3(t)\)

When the system fails in \((t_k, t_k^*)\), the relevant cost is \((C_f + (i + k - 1) C_s)\), including the cost of \((i + k - 1)\)th inspection and failure replacement cost. But if the severe defective stage is identified at \(t_k^*\), the relevant cost is \((C_p + (i + k) C_s)\), including the cost of \((i + k)\)th inspection and maintenance cost. The relevant cost of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle cost \(EC_3(t)\) can be derived as

\[
EC_3(t)
\]

\[
= \sum_{i=1}^{\infty} \left( (C_f + (i + k - 1) C_s) P(t_k < T_f < t_k^*)
\]

\[
+ (C_p + (i + k) C_s) P(T_p = t_k^*) \right).
\]

(13)

### 3.4.2. Expected Renewal Cycle Length \(EL_3(t)\)

The relevant length of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle length \(EL_3(t)\) can be derived as

\[
EL_3(t)
\]

\[
= \sum_{i=1}^{\infty} \left( \int_{(i-1)t}^{it} f_U(u) \right.
\]

\[
\cdot \left( \int_{t_k-u}^{t_k^*-u} f_V(v) f_H(t_k^* - u - v) \, dv \, du \right).
\]

(14)
3.4.3. Expected Cost per Unit Time $C(t)$. Based on the derived expected cycle cost and length, expected cost per unit time can be calculated using the same way as (10)

$$ C(t) = \frac{EC(t)}{EL(t)} = \frac{EC_1(t) + EC_3(t)}{EL_1(t) + EL_3(t)}. \quad (15) $$

4. Cost Models with Imperfect Inspection

Imperfect inspection means that inspections may not identify the minor defective stage perfectly but can identify the severe defective stage perfectly. Two different cost models with imperfect inspection are given in this section to optimize the inspection intervals. Similarly, two renewal scenarios are considered here and the renewal probabilities should be formulated. According to the different decisions when identifying the minor defective stage, two optimization models are proposed.

**Model 1.** The inspection interval is shortened only when the minor defective stage is identified at the first time.

4.1. The System Is Renewed before the Minor Defective Stage Is Identified, as Shown in Figure 4. There may exist two renewal scenarios.

(1) The system is renewed when it fails in $((i-1)t, it)$. The normal stage ends and then the minor defective stage starts within $((j-1)t, jt)$. The length of minor defective stage is longer than $(i-1)t - u$ but all the inspections in $(jt, (i-1)t)$ miss the minor defect because of the imperfect inspection. The probability is given as

$$ P((i-1)t < T_f < it) = P((j-1)t < U < jt, (i-1)t - U < V < it - U, 0 < H < it - U - V) $$

$$ = \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{\delta((i-1)t-u)}^{\delta(i-1)t-u} (1-r)^{i-j} f_U(u) f_V(v) F_H(it - u - v) du dv, \quad (16) $$

where we define $\delta(u) = \max(u, 0)$.

(2) The system is renewed at $it$ as the severe defective stage is identified by an inspection. The minor defective stage starts within $((j-1)t, jt)$ and ends within $((i-1)t, it)$. Of course, all the inspections within $(jt, (i-1)t)$ miss the minor defective stage. The probability is

$$ P(T_p = it) = P((j-1)t < U < jt, (i-1)t - U < V < it - U, H > it - U - V) $$

$$ = \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{\delta((i-1)t-u)}^{\delta((j-1)t-u)} (1-r)^{i-j} f_U(u) f_V(v) F_H(it - u - v) du dv $$

$$ + (C_p + iC_s) \int_{(j-1)t}^{jt} \int_{\delta((i-1)t-u)}^{\delta((j-1)t-u)} (1-r)^{i-j} f_U(u) f_V(v) F_H(it - u - v) du dv, \quad (17) $$

Consequently, the expected renewal cycle cost and length with different renewal scenarios can be derived.

4.1.1. Expected Renewal Cycle Cost $EC_{11}(t)$. When the system fails in $((i-1)t, it)$, the relevant maintenance cost is $(i-1)C_s + C_f$. However, once the severe defective stage is identified at $it$ by an inspection, the relevant cost is $(iC_s + C_p)$. Therefore, the expected renewal cycle cost $EC_{11}(t)$ can be derived through multiplying the relevant cost of each event by the renewal probability, respectively;

$$ EC_{11}(t) = \sum_{i=1}^{\infty} \left( (C_f + (i-1)C_s) P((i-1)t < T_f < it) + (C_p + iC_s) P(T_p = it) \right) $$

$$ = \sum_{i=1}^{\infty} \left( (C_f + (i-1)C_s) \right. $$

$$ \cdot \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{\delta((i-1)t-u)}^{\delta((j-1)t-u)} (1-r)^{i-j} f_U(u) f_V(v) F_H(it - u - v) du dv $$

$$ + \left. (C_p + iC_s) \int_{(j-1)t}^{jt} \int_{\delta((i-1)t-u)}^{\delta((j-1)t-u)} (1-r)^{i-j} f_U(u) f_V(v) F_H(it - u - v) du dv \right). \quad (18) $$

4.1.2. Expected Renewal Cycle Length $EL_{11}(t)$. The relevant length of each event multiplies by the renewal probability,
Failure renewal
Inspection renewal
Failure renewal

**Figure 5:** The system is renewed after minor defective stage is identified at the first time with imperfect inspection.

respectively; thus the expected renewal cycle length $EL_{11}(t)$ can be derived as

$$EL_{11}(t) = \sum_{i=1}^{\infty} \left( \int_{0}^{(i-1)t+h} ((i-1)t+h) \cdot P((i-1)t+h < T_f < (i-1)t+h+dh) + itP(T_p = it) \right)$$

$$= \sum_{i=1}^{\infty} \left( \int_{0}^{(i-1)t+h} ((i-1)t+h) \right)$$

$$\cdot \sum_{j=1}^{i} \int_{jt}^{(j-1)t} \int_{0}^{(i-1)t+u} (1-r)^{i-j} \cdot f_U(u) f_V(v) F_H(it+nt/2-u-v) \cdot \left(1-F_H(it+nt/2-u-v)\right) dv du dh.$$

4.2. The System Is Renewed after the Minor Defective Stage Is Identified at it, as Shown in Figure 5. There also exist two renewal scenarios.

(1) The system fails in $(it + (n-1)t/2, it + nt/2)$ after the minor defective stage is identified at it at the first time. Inspections in $(jt, (i-1)t)$ all miss the minor defective stage. The minor defective stage ends within $(it + (n-1)t/2, it + nt/2)$ and then severe defective stage and failure happens in the same interval. The probability of such a renewal is

$$P\left(it + \frac{(n-1)t}{2} < T_f < it + \frac{nt}{2}\right)$$

$$= P\left((j-1)t < U < jt, it + \frac{(n-1)t}{2} - U < V < it + \frac{nt}{2} - U, 0 < H < it + \frac{nt}{2} - U - V\right)$$

$$= \sum_{j=1}^{i} \int_{jt}^{j-1} \int_{it+nt/2-u}^{it+(n-1)t/2-u} (1-r)^{i-j}$$

$$\cdot r f_U(u) f_V(v) F_H(it+nt/2-u-v) dv du. \quad (19)$$

(2) The minor defective stage of system starts within $(jt, (j-1)t)$ and is identified at it at the first time by an inspection. Then the subsequent inspection interval is halved. The length of severe defective stage is longer than $(it + nt/2 - u - v)$ and is identified at it + nt/2. The probability of such a renewal is

$$P\left(T_p = it + \frac{nt}{2}\right)$$

$$= P\left((j-1)t < U < jt, it + \frac{(n-1)t}{2} - U < V < it + \frac{nt}{2} - U, H > it + \frac{nt}{2} - U - V\right)$$

$$= \sum_{j=1}^{i} \int_{jt}^{j} \int_{it+nt/2-u}^{it+(n-1)t/2-u} (1-r)^{i-j}$$

$$\cdot r f_U(u) f_V(v)\left(1-F_H(it+nt/2-u-v)\right) dv du.$$

Accordingly, the expected renewal cycle cost and length with different renewal scenarios can be derived as follows.
4.2.1. Expected Renewal Cycle Cost $EC_{22}(t)$. The expected renewal cycle cost $EC_{22}(t)$ can be derived through multiplying the relevant cost of each event by the renewal probability; respectively,

$$EC_{22}(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( (C_f + (i + n - 1) C_s) \cdot P(\frac{it}{2} + \frac{(n-1)t}{2} < T_f < it + nt) + (C_p + (i + n) C_s) P(T_p = it + \frac{nt}{2}) \right)$$

$$= \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( (C_f + (i + n - 1) C_s) \cdot \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{\frac{it}{2} + \frac{(n-1)t}{2} + u}^{jt} (1-r)^{i-j} \cdot rf_U(u) f_V(v) F_H \cdot \left( it + \frac{nt}{2} + u - v \right) dV du \right)$$

$$+ \left( C_p + (i + n) C_s \right) \cdot \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{\frac{it}{2} + \frac{(n-1)t}{2} + u}^{it} (1-F_H) \cdot \left( it + \frac{nt}{2} + u - v \right) dV du \right) \cdot \left( \frac{h}{2} \right).$$

(22)

4.2.2. Expected Renewal Cycle Length $EL_{22}(t)$. The relevant length of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle length $EL_{22}(t)$ can be formulated as

$$EL_{22}(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( \int_{0}^{\frac{t}{2}} \left( it + \frac{(n-1)t}{2} + h \right) P\left( \frac{it}{2} + \frac{(n-1)t}{2} + h < T_f < it + \frac{nt}{2} \right) + \frac{(n-1)t}{2} + h + dh \right)$$

$$+ \left( \frac{nt}{2} \right) P\left(T_p = it + \frac{nt}{2} \right).$$

(23)

4.2.3. Expected Cost per Unit Time $C(t)$. Therefore, based on the derived expected cycle cost and length, expected cost per unit time can be calculated:

$$C(t) = \frac{EC(t)}{EL(t)} = \frac{EC_{11}(t) + EC_{22}(t)}{EL_{11}(t) + EL_{22}(t)}.$$ 

(24)

Model 2. It is shortening the inspection interval if only the minor defective stage is identified.

4.3. The System Is Renewed before Identifying the Minor Defective Stage with Imperfect Inspection. This scenario is the same as Section 4.1.

4.4. The System Is Renewed after the Minor Defective Stage Is Identified. As this model supposes that if only the minor defective stage is identified, the inspection interval is halved. The minor defective stage may be missed by the inspections because of the imperfect inspection. Moreover, the inspection may miss the minor defect between two points of identifying the minor defect. The modeling process is complex. Therefore, the scenario that system renews after identifying the minor defective stage at the third time is present as an example. The derivation of other scenarios follows the same principle.
The system is renewed after identifying the minor defective stage at the third time, as depicted in Figure 6.

(1) A failure renewal happens when the system deteriorates into the minor defective stage within \( ((j - 1)t, jt) \) and fails in \( (tk, t^*k) \). The minor defective stage is identified three times in \((jt, tk)\). The probability of such a failure renewal is

\[
P(t_k < T_f < t^*_k) = P\left( (j - 1)t < U < jt, \ tk - U < V < t^*_k - U, \ 0 < H < t^*_k - U - V \right)
\]

\[
= \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{t_k-u}^{t^*_k-u} (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot F_H(t^*_k - u - v) \cdot dvdu.
\]

(2) An inspection renewal happens when the severe defective stage of system is identified at \( t^*_k \). The minor defective stage lasts from \( jt \) to \( tk \) and is identified by inspections three times. So the subsequent inspection interval is \( t/8 \). The probability of this inspection renewal is

\[
P(T_p = t^*_k) = P\left( (j - 1)t < U < jt, \ tk - U < V < t^*_k - U, \ H > t^*_k - U - V \right)
\]

\[
= \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{t_k-u}^{t^*_k-u} (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot (1 - F_H(t^*_k - u - v)) \cdot dvdu.
\]

4.4.1. Expected Renewal Cycle Cost \( EC_{33}(t) \). The expected renewal cycle cost \( EC_{33}(t) \) can be derived through multiplying the relevant cost of each event by the renewal probability, respectively;

\[
EC_{33}(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( (C_f + (i+n+k)C_s) P(t_k < T_f < t^*_k) + (C_p + (i+n+k+1)C_s) P(T_p = t^*_k) \right)
\]

\[
= \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( (C_f + (i+n+k)C_s) \cdot \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{t_k-u}^{t^*_k-u} (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot F_H(t^*_k - u - v) \cdot dvdu \right.

\]

\[
+ \left. (C_p + (i+n+k+1)C_s) \cdot \sum_{j=1}^{i} \int_{(j-1)t}^{jt} \int_{t_k-u}^{t^*_k-u} (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot (1 - F_H(t^*_k - u - v)) \cdot dvdu \right).
\]

(27)

4.4.2. Expected Renewal Cycle Length \( EL_{33}(t) \). The relevant length of each event multiplies by the renewal probability, respectively; thus the expected renewal cycle length \( EL_{33}(t) \) can be formulated as

\[
EL_{33}(t) = \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \int_{0}^{t/8} (t_k + h) P(t_k + h < T_f < t_k + h + dh) + t^*_k P(T_p = t^*_k)
\]

\[
= \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( \int_{0}^{t/8} (t_k + h) \right)

\]

\[
= \sum_{i=1}^{\infty} \sum_{n=1}^{\infty} \left( \int_{0}^{t/8} (t_k + h) \right)

\]

\[
\cdot \int_{t_k-u}^{t^*_k-u} (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot (1 - F_H(t^*_k - u - v)) \cdot dvdu.
\]

(27)
\[ r^3 f_U(u) f_V(v) f_H \cdot (t_k + h - u - v) dvdu dh \]
\[ + t_k^i \int_{j}^{j} t_k^j (1 - r)^{(i-j+n+k-2)} \cdot r^3 f_U(u) f_V(v) \cdot (1 - F_H(t_k^* - u - v)) dvdu \].

\[(28)\]

4.4.3. Expected Cost per Unit Time $C(t)$. Based on the derived expected cycle cost and length, expected cost per unit time can be formulated:

\[ C(t) = \frac{EC(t)}{EL(t)} = \frac{EC_{11}(t) + EC_{33}(t)}{EL_{11}(t) + EL_{33}(t)}. \]

\[(29)\]

5. Numerical Example

A numerical example is presented in this section to compare the superiority of two models according to the different measures when identifying the minor defective stage, thus presenting a scientific maintenance decision for safety assessment of NPPs. In order to avoid the influence of relevant parameters, two sets of parameters are set up here corresponding with the perfect inspection and imperfect inspection, respectively. These three stages of failure process are assumed to follow Weibull distribution because of the perfect performance of Weibull distribution in describing the lifetime of system, as shown in

\[ f_{X_n}(x) = a_n b_n (a_n x)^{b_n-1} e^{-(a_n x)^{b_n}}, \]

\[(30)\]

where $a_n, b_n$ are scale parameter and shape parameter, respectively.

Example 1 (perfect inspection). As the total lifetime experiment of systems or equipment following to three-stage failure process is hard to implement, we cannot possess the complete experimental data at present. So the distribution and cost parameters are assumed in Table 1.

According to the given parameters and cost models which have been built in Sections 3 and 4, we can derive the outputs of proposed models and determine the optimal inspection interval through MATLAB programming. The results of models 1 and 2 with perfect inspection are shown in Figures 7 and 8, respectively.

In order to make comparison for the two different measures when identifying the minor defective stage, the results of the optimal injection interval $t^*$ are shown in Table 2. We can see that the expected cost per unit time of model 1 is smaller than model 2 no matter what the value of $k$ is. Moreover, with the increasing of $k$ in model 2, the maintenance cost is larger. The relevant expected cost values of models 1 and 2 are given in Table 2.

From Table 2, it can be seen that the optimal expected cost per unit time $C(t)$ of model 1 is 8.4007 and the optimal inspection interval $t^*$ is 47. However, the optimal $C(t)$ is 9.6727 when the optimal $k^*$ is 2 and optimal $t^*$ is 42. Therefore, model 1 can

---

**Table 1: The distribution and cost parameters.**

<table>
<thead>
<tr>
<th>Distribution parameters</th>
<th>Cost parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_s$ $C_p$ $C_f$</td>
<td>$U$ $V$ $H$</td>
</tr>
<tr>
<td>$a_1$ $a_2$ $a_3$ $b_1$ $b_2$ $b_3$</td>
<td></td>
</tr>
<tr>
<td>80 800 5000 0.0056 1.25 0.026 0.68 0.021 2.56</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: The optimal results of models 1 and 2 with perfect inspection.**

<table>
<thead>
<tr>
<th>$k$</th>
<th>$t^*$</th>
<th>Model 1 $C(t)$</th>
<th>Model 2 $C(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>42</td>
<td>9.6727</td>
<td>9.6727</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>9.8194</td>
<td>9.8194</td>
</tr>
<tr>
<td>4</td>
<td>41</td>
<td>9.8781</td>
<td>9.8781</td>
</tr>
<tr>
<td>5</td>
<td>41</td>
<td>9.9010</td>
<td>9.9010</td>
</tr>
<tr>
<td>6</td>
<td>41</td>
<td>9.9093</td>
<td>9.9093</td>
</tr>
<tr>
<td>7</td>
<td>41</td>
<td>9.9118</td>
<td>9.9118</td>
</tr>
</tbody>
</table>
save cost 13.15% more than model 2. So model 1 is optimal compared with model 2 conditioning on the cost parameters. In other words, the measure that shortening the inspection interval only when identifying the minor defective stage at the first time can save more maintenance cost. Although the action of model 2 can prevent the failure of system definitely, frequent inspection maintenance will increase the maintenance cost. Therefore, the optimal maintenance decision is the action that shortening the inspection interval only when the minor defective stage is identified at the first time, which is also beneficial for making up the maintenance plan and improving the reliability and safety of NPPs.

Example 2 (imperfect inspection). In this section, we give another set of parameters to avoid the influence of different parameters. The distribution and cost parameters are shown in Table 3. The probability of imperfect inspection $r$ is 0.6.

According to the given distribution and cost parameters, the outputs of models 1 and 2 are shown in Figure 10. The relevant expected cost values of models 1 and 2 are given in Table 4. With imperfect inspection case, model 1 can save maintenance cost 12.4% more than model 2 with condition 1.

From Figure 10 and Table 4, we can derive the same conclusion as Example 1 that model 1 is more optimal than model 2.

According to the two examples in this section, we can derive the conclusion that shortening the inspection interval only when the minor defective stage is identified at the first time is optimal no matter the inspection is perfect or imperfect. Moreover, this conclusion can help maintenance managers make scientific decisions, which will also be helpful for safety assessment of NPPs. Of course, the results cannot be generated to all cases as the output result of proposed models depends on the model parameters, particularly the cost parameters.

### Table 3: The distribution and cost parameters.

<table>
<thead>
<tr>
<th>Cost parameters</th>
<th>Distribution parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_s$</td>
<td>$C_p$</td>
</tr>
<tr>
<td>50</td>
<td>200</td>
</tr>
</tbody>
</table>

### Table 4: The optimal results of models 1 and 2 with imperfect inspection.

<table>
<thead>
<tr>
<th></th>
<th>Model 2</th>
<th></th>
<th>Model 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t^*$</td>
<td>$C(t)$</td>
<td>$t^*$</td>
<td>$C(t)$</td>
</tr>
<tr>
<td>Case 1</td>
<td>16</td>
<td>20.4592</td>
<td></td>
</tr>
<tr>
<td>Case 2</td>
<td>16</td>
<td>20.1346</td>
<td>13</td>
</tr>
<tr>
<td>Case 3</td>
<td>15</td>
<td>20.3175</td>
<td></td>
</tr>
</tbody>
</table>

6. Conclusions

As evaluating the system reliability is one of the important tasks of PSA for NPPs and maintenance is considered as one of the most important factors for affecting the total reliability and safety, an enhanced preventive maintenance optimization model based on a three-stage failure process with perfect and imperfect inspection cases is proposed to optimize the inspection interval of NPPs by minimizing the expected cost per unit time. The lifetime of system before failure is divided into three stages, namely, normal, minor defective, and severe defective stages, which is closer to reality and corresponds to the three-color scheme commonly used in industry. When the minor defective stage is identified, two measures are considered for comparison: one is that halving the inspection interval only when the minor defective stage is identified at the first time; the other one is that if only identifying the minor defective stage, the subsequent inspection interval is halved. Maintenance is implemented immediately once the severe defective stage is identified. The results of the numerical example show that the measure that halving the inspection interval when the minor defective stage is identified at the first time can save more maintenance cost no matter the inspection is perfect or imperfect.
The optimal inspection interval can be found through the proposed model. The conclusions can also help the managers make scientific maintenance decisions and benefit the safety assessment of nuclear power plants.

The further relevant research can be developed as (1) joint optimization policy of inspection and spare parts based on the three-stage failure process, (2) considering the maintenance window at the defective stages, and (3) case studies that should be considered. These points will be researched in the future.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgment

The project was supported by the Natural Science Foundation of Hebei Province (E2015506012).

References

In case of some nuclear power plants constructed at the soft soil sites, liquefaction should be analysed as beyond design basis hazard. The aim of the analysis is to define the postevent condition of the plant, definition of plant vulnerabilities, and identification of the necessary measures for accident management. In the paper, the methodology of the analysis of liquefaction effects for nuclear power plants is outlined. The procedure includes identification of the scope of the safety analysis and the acceptable limit cases for plant structures having different role from accident management point of view. Considerations are made for identification of dominating effects of liquefaction. The possibility of the decoupling of the analysis of liquefaction effects from the analysis of vibratory ground motion is discussed. It is shown in the paper that the practicable empirical methods for definition of liquefaction susceptibility provide rather controversial results. Selection of method for assessment of soil behaviour that affects the integrity of structures requires specific considerations. The case of nuclear power plant at Paks, Hungary, is used as an example for demonstration of practical importance of the presented results and considerations.

1. Introduction

Proper understanding and assessment of safety of nuclear power plants (NPPs) with respect to external hazards became very important after 11 March 2011. Experience of Niigata-ken Chuetsu-oki earthquake of 16 July 2007, the $M_w \approx 5.8$ Mineral (Virginia) earthquake of 2011, and also the response of Japan nuclear power plants to the Great Tohoku earthquake demonstrated that the design practice ensures the safety of nuclear power plants with respect to the vibratory ground motion. However, secondary effects of earthquakes that have not been properly considered in the design can heavily damage the plants, as in the case of the Fukushima Dai-ichi plant, where the tsunami led to fatal consequences after the plant survived the beyond design base ground vibratory motions.

Soil liquefaction can also be one of those secondary effects of earthquakes that should be accounted for at soft soil sites. Usually the liquefaction is not considered as a design base hazard. If the soil at the site is susceptible to the liquefaction, soil improvement and appropriate foundation design have to be applied for excluding the potential hazard. However, at some NPP sites, soil liquefaction has to be considered as a beyond design basis event, especially if the safety factor to liquefaction is rather low in case of design base earthquake.

Paks NPP is the only nuclear power plant in Hungary providing more than 40% of domestic electricity production. Originally, the plant was not designed for earthquake since the site seismic hazard was underestimated and the former Soviet design requirements did not require specific design measures for this case. In the early nineties, the site seismic hazard has been reevaluated using comprehensive probabilistic seismic hazard assessment (PSHA) methodology [1]; new seismic design basis has been defined with peak ground acceleration 0.25 g for 10⁻⁴/a nonexceedance level. Extensive safety upgrading measures have been implemented to comply with new design basis requirements [2]. The Paks site soil conditions are shown in Table 1. The groundwater level is about 8.5 m below grade and varies with the seasonal
variation of the Danube water level. Probabilistic liquefaction hazard analysis performed in the early nineties [3] provided annual probability for liquefaction less than $10^{-4}$/a. Consequently, the liquefaction was not considered as a design basis hazard, since the $10^{-4}$/a annual frequency is the criteria for accounting for an external hazard in the design basis.

After implementing the seismic upgrading measures, comprehensive seismic probabilistic safety assessment (seismic PSA) has also been performed. The seismic PSA demonstrates significant margins with respect to earthquake vibratory effect. The seismic PSA also accounted for the possibility of liquefaction via rather simplified way, practically assuming “cliff-edge effect” when the soil liquefies. The seismic PSA has shown that the liquefaction could be one of the essential contributors to the core damage. This finding motivated the investigations of the liquefaction hazard and safety consequences of the liquefaction. These efforts received high attention after Fukushima accident. Recently the liquefaction hazard as well as the plant response to liquefaction is extensively investigated for definition of safety margins and development of severe accident management procedures and measures. These activities are part of the national programme developed in the frame of focused safety assessment (stress-test) initiated by the European Union [4].

Conclusiveness of the beyond design base safety analysis for liquefaction depends on proper consideration of epistemic and aleatory uncertainties related to the uncertainties of assessment of ultimate behaviour of plant structures due to low probability complex effects. Therefore, the beyond design basis safety assessment for liquefaction is a complex procedure that integrates deterministic as well as probabilistic elements. The procedure consists of the following tasks:

(i) selection/development of the method for analysis of plant response to liquefaction that includes identification of accident scenarios and selection of methods for analysis of safety relevant systems, structures, and components,

(ii) characterisation of liquefaction hazard that includes

   (a) probabilistic seismic hazard assessment,

   (b) investigation of the soil properties,

   (c) selection of the appropriate methods for characterisation of liquefaction hazard,

   (d) calculation of the liquefaction effects relevant for evaluation of plant response,

(iii) performing the safety analysis, identification of the plants vulnerabilities, and definition of accident mitigation measures.

Although the methods for assessment of the liquefaction hazard as well as the deterministic and probabilistic methodologies for evaluation of the liquefaction consequences have been widely studied, there is no experience or precedence for performing full scope safety analysis of an operating nuclear power plant for liquefaction.

In the paper, the issues of the selection of the method for beyond design basis safety analysis, practical problems of the selection of the method for assessment of the liquefaction hazard, and the calculation of relevant liquefaction effects are presented and discussed.

2. Selection of Safety Analysis Methodology

Recently several methodical documents have also been published on the development of severe accident management procedures, for example, [5]. According to this, plant vulnerabilities in the case of accidents beyond the design basis should be identified; knowledge on the behaviour of the plant during a beyond design basis accident should be obtained; the phenomena that may occur and their expected timing and severity should be identified. However, unified methodology does not exist for analysis of beyond design basis accidents caused by external events in combination with secondary effects, for example, earthquake and earthquake-induced soil
Table 2: Relation between mechanism of structural displacement and earthquake parameters as well as parameters of the structure.

<table>
<thead>
<tr>
<th>Increase in parameter</th>
<th>Localization</th>
<th>Sedimentation</th>
<th>Consolidation due to excess pore pressure dissipation</th>
<th>Partial bearing failure due to strength loss in foundation soil</th>
<th>SSI-induced building ratcheting due to cyclic foundation loading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak ground acceleration (PGA)</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
</tr>
<tr>
<td>Liquefiable layer rel. density (Dr)</td>
<td>↓↓</td>
<td>↓↓</td>
<td>↓</td>
<td>↓↓</td>
<td>↑↓</td>
</tr>
<tr>
<td>Liquefiable layer thickness</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↓</td>
</tr>
<tr>
<td>Foundation width</td>
<td>↓</td>
<td>↑↑</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Static shear stress ratio, $\tau_{\text{static}}/\sigma'_{\text{o}}$</td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
<td>↑↓</td>
<td>—</td>
</tr>
<tr>
<td>Height/width ratio of structure</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>—</td>
<td>↑↑</td>
</tr>
<tr>
<td>Building weight</td>
<td>↑↓</td>
<td>↑↓</td>
<td>↑↓</td>
<td>↑↓</td>
<td>↑↑</td>
</tr>
<tr>
<td>3D drainage</td>
<td>↑↑</td>
<td>↓</td>
<td>↑↑</td>
<td>↓</td>
<td>↑↑</td>
</tr>
</tbody>
</table>

Liquefaction. Contrary to this, essential progress has been achieved in the area of analysis for earthquake plus tsunami in the frame research programme of the International Atomic Energy Agency International Seismic Safety Centre [6]. The difficulties relate to the assessment of superposed effects of the ground vibratory motion and the liquefaction; that is, the liquefaction affects the plant twofold.

(i) Due to liquefaction, the site response becomes strongly nonlinear; that is, the liquefaction affects promptly the ground vibratory motion.

(ii) Liquefaction causes soil settlement, lateral spread, and so forth that can damage the plant structures.

These consequences of liquefaction are caused by several mechanisms of soil deformations that depend on the soil conditions, earthquake parameters, and parameters of the structure in a very complex manner. This is shown in Table 2 (see also [7]).

In Table 2, the arrows up indicate an increasing effect with increasing parameter value, while the arrows down indicate decreasing effect. Doubling of arrows corresponds to strong effect. Arrows in both directions indicate that the increasing of a parameter can cause controversial effects depending on conditions.

For the illustration of the plant response, let us take the simplified plant event-tree of the earthquake and subsequent liquefaction event shown in Figure 1. Loss of offsite power (LOSP) is assumed to be the initiating event caused by earthquake vibratory motion. The reactor shutdown system (denoted by $A$) shall ensure the subcriticality. The emergency power system ($B$) and the emergency core cooling system ($C$) are needed for avoiding the core damage. The success path after earthquake will be as follows: the reactor is subcritical; the emergency power supply and the emergency core cooling are ensured.

In case of Paks NPP, analysis of the site soil conditions and the features of critical plant structures resulted in the conclusion that the differential ground settlement due to liquefaction is the dominating effect that can damage the critical plant structures. The differential settlement can result in tilting and relative displacement between adjacent buildings that result in loss of integrity. The relative displacement between buildings and underground piping and cables can damage these communication lines. The latter affects mainly the power cables of the emergency power supply and the piping to the ultimate heat sink. It means the emergency power supply ($B$) and the systems for heat removal ($C$) could be affected by the soil settlement with time delay $\Delta t$ after strong motion starts. Once worked, the system $A$ will continue to ensure the subcriticality, though the reactor might be tilted together with the reactor building due to liquefaction. Although it is not indicated in Figure 1, the differential settlement can also cause loss of integrity of the containment that should be also evaluated. Assuming the above scenario, the liquefaction can be considered as a separate load case subsequent to the vibratory motion.

The analysis of safety consequences of the above scenario can be performed by either deterministic or probabilistic method.

A probabilistic safety analysis of liquefaction (liquefaction PSA) requires the characterisation of the hazard, development of the plant event-trees and fault-trees, and knowledge of the fragility of systems, structures, and components (SSCs) relevant to safety. The fragility has to be defined as a function of engineering demand parameter that should be correlated with appropriate intensity measure characterising the earthquake hazard.

A performance-based earthquake engineering (PBEE) probabilistic framework for evaluation of the risk associated with liquefaction has been developed in [8–10]. In the PBEE, the earthquake is characterized by an intensity measure (IM),
for example, peak ground acceleration. An engineering demand parameter (EDP) has to be identified and the EDP should be correlated with the damage measures (DM) that are measures of physical effect of EDP. The risk associated with DM has to be expressed in some decision variables (DV) applicable for the risk characterization (some measure of loss). The mean annual rate of exceedance of a given DV level can be calculated if the annual rate of the IM and the conditional probabilities connecting the IM to EDP, the EDP to DM, and the DM to DV are known. The mean annual rate of exceedance of a given DV level, \( \lambda_{DV} \), can be expressed as

\[
\lambda_{DV} = \sum_{k=1}^{N_{DM}} \sum_{j=1}^{N_{EDP}} \sum_{i=1}^{N_{IM}} P_{[\text{DV} \mid \text{DM}_k]} \cdot P_{[\text{DM}_k \mid \text{EDP}_j]} P_{[\text{EDP}_j \mid \text{IM}_i]} \Delta \lambda_{IM,i}
\]

(1)

where functions of the type \( P\{a = a' \mid b = b'\} \) describe the conditional probability of random variable, a given \( b = b' \). The \( \Delta \lambda_{IM,i} \) is the \( i \)th increment of mean annual rate of exceedance of IM. The \( N_{DM}, N_{EDP}, \) and \( N_{IM} \) are the numbers of increments corresponding to DM, EDP, and IM, respectively. In case of nuclear power plant safety analysis, the DV can be associated with loss of safety function if the measure of damage is exceeding the level of DM. The aggregate mean annual frequency of exceeding a particular value of DV is then determined by summing up the contributions from all combinations of possible intensity measures, engineering demand parameters, and damage measures.

In spite of the comprehensiveness of the PBEE method, it is rather difficult to apply, since identification of parameters DM, EDP, and IM and definition of \( \lambda_{IM,i} \) are not trivial. The calculation of the aggregate mean annual frequency for DV requires the knowledge of multivariate distribution of intensity measures; hence the engineering demand parameters like soil settlement depend on the peak ground acceleration and on the magnitude of earthquake. The calculation can be simplified, if the marginal distributions of intensity measures are used which implies week correlation between the intensity measures. Further difficulties are related to the definition of the conditional probabilities (fragility functions) of the plant SSCs. The available information on the fragility of buildings, underground structures, and lifelines of nuclear power plants is rather scarce. Nevertheless, a liquefaction PSA has certain
advantages, since it quantifies the core damage and early large release frequencies and identifies the plant vulnerabilities. However, the liquefaction PSA would not provide input information for the design of upgrading or mitigating measures.

Considering the abovementioned difficulties, a deterministic approach has been adopted for Paks NPP for the beyond design basis safety analysis of the liquefaction.

Deterministic safety analysis assesses the integrity and function of the plant SSCs while calculating the loads due to liquefaction, the stresses, and strains caused by these loads and comparing these to ultimate values. Thus, the procedure for analysis of plant response to the liquefaction as beyond design base event consists of the following steps.

(1) The first is probabilistic seismic hazard assessment that provides the peak ground acceleration and deaggregation matrices that are used in computations of magnitude for the liquefaction hazard analysis.

(2) The second is calculation of soil settlements due to the liquefaction.

(3) The third is identification of SSCs within the scope of liquefaction safety analysis. These are the SSCs needed for ensuring the heat removal from reactor and spent fuel pool as well as the SSCs that are important for accident management. The identification of the SSCs has been done in the frame of Targeted Safety Review (see [4]). Thus, the SSCs within the scope of liquefaction safety analysis are as follows.

(a) SSCs have to be functional or preserve their integrity, as it is required for emergency heat removal (see [4, 5]). In the case of Paks NPP that are first of all the essential service water system and the emergency power supply system, the essential service water system consists of piping, water intake structures, and water intake control building. The underground pipelines connect the pumps located in the water intake building to the main reactor building and diesel building while crossing the lower level in the turbine hall. There are also back-up systems (e.g., the fire water system) that can be used as ultimate heat sinks in case of severe accident and back-up power supply systems, too. These should be also included in the scope of analysis.

(b) Containment function has to be ensured for limiting the radioactive releases.

(c) Structures and systems with limited radioactive inventory, for example, auxiliary building, should preserve certain level of structural integrity for limiting the site releases.

(d) Control rooms and the structures along the escape routes: integrity and habitability of the barrack of fire brigade and Protected Command Centre have to be analysed and ensured.

(e) The Laboratory and Service Building that is connected to the controlled area of the plant has to be checked, whether the life safety is ensured and the escape routes are safe.

(f) Buildings that may collapse should not damage the essential service water system and emergency power lines or hinder the implementation of emergency measures.

(4) The fourth is definition of the criteria for assessing whether the SSCs identified above comply with the above requirements as well as the methods for structural analysis. Definition of assumptions applicable for material properties and load combinations are also included in this step of process. Examples are as follows.

(a) Permanent deformation of pipelines of the essential service water systems can be accepted assuming that the overall integrity and leak-tightness are ensured.

(b) According to IAEA Safety Guide NS-G-1.10 [11], the following conditions can be accepted for the containment regarding structural integrity.

(i) Level II: local permanent deformations are possible. Structural integrity is ensured, though with margins smaller than those for design base.

(ii) Level III: significant permanent deformations are possible, and some local damage is also expected. Normally, this level is not considered in case of severe accidents.

For leak-tightness, the following levels could be considered.

(i) Level II: the leak rate may exceed the design value, but the leak-tightness can be adequately estimated and considered in the design.

(ii) Level III: leak-tightness cannot be ensured owing to large deformations of the containment structure. Structural integrity may still be ensured.

Considering the design of Paks NPP, large permanent deformations of the containment walls and floors are allowed when the deformations are within the strain limits allowable for the liner that ensures the necessary leak-tightness of the containment.

Relative displacement between containment and structures connected to the containment has to be assessed from the point of view of integrity of essential service water pipelines crossing these locations.

(c) In case of Laboratory and Service Building, near collapse conditions (according to EUROCODE 8 Part 3 or FEMA-356 2000) are allowed, while the evacuation is ensured via safe escape routes. A near collapse condition is also acceptable in case of auxiliary building.

(d) Specific attention has to be paid to the water intake structure whether the functioning of
the pumps and free cross section for intake is ensured.

Best estimate models, mean values of loads and material properties can be used in the analysis of the liquefaction effects. In best estimate models, contribution to the resistance of nonstructural elements can be accounted for. The calculation can be linear or nonlinear static. In case of containment (main reactor building), coupled soil-structure model is applicable.

The structures within the scope of safety analysis are rather different. The main reactor building foundation is at the depth of 8.5 m, below groundwater table, while the piping of essential service water system is located near surface in dry sand. The analysis methods selected for each structure within the scope have to fit to the specific design and soil conditions.

(5) The fifth is performing the analysis, conclusion on the plant response to liquefaction, and identification of the safety upgrading measures for ensuring the effective accident management and development of accident management guidance.

3. Characterisation Liquefaction Hazard

3.1. Geotechnical Investigations. Comprehensive geotechnical survey has been made for better understanding the site conditions and updating the database obtained prior to the construction of the plant. Altogether at the site (500 m × 100 m area), there are nearly 500 boreholes and other test points and more than 100 groundwater-monitoring wells. Site geotechnical survey includes mapping soil stratigraphy, in situ definition of soil properties, full scope laboratory testing of samples, cyclic triaxial and resonant column tests, Standard Penetration Tests (SPT), Cone Penetration Tests (CPT), Piezometric Cone Penetration Tests (CPTu), and Seismic Cone Penetration Tests (SCPT). Data are stored and presented in GeoDin database. The geotechnical investigations were performed in compliance with standards ISO 22475, ISO 22476, ISO/TS 17892, ASTM D3999-11, and ASTM 4015-07.

It has to be noted that the performance of geotechnical investigations at the plant site was rather difficult because of underground structures and lifelines. On the other hand, the soil conditions close to the buildings were disturbed due to foundation excavations. Therefore, a control area has been selected north to the plant where undisturbed soil conditions could be studied. Of course the soil conditions at the control area are not completely identical to those below critical plant structures. These differences have to be accounted for properly.

The classical method for determining liquefaction potential is based on SPT measurements, which had been the most widely used procedure. CPT, however, has approached the same level, and newly developed CPT based correlations now represent coequal or even better status with regard to accuracy and reliability. Compared to SPT, CPT offers advantages with regard to cost, efficiency, repeatability, and consistency. The accuracy of SPT measurements is operator-dependent and their usefulness depends on the soil type: they give the most useful results in case of fine-grained sands, while in case of clays and gravelly soils they provided results, which may very poorly represent the true soil conditions. However, the most important aspect is the continuity of data over depth. SPT can only be performed at vertical spacing of about 75 cm or more, so it can completely miss thin (but potentially important) liquefiable strata. CPT, in contrast, is fully continuous and so “misses” nothing.

3.2. Analysis of the Liquefaction Potential. In practice, empirical methods based on in situ geotechnical tests are the most frequently used for liquefaction potential evaluation. For the Paks site, the preliminary calculations using well-known empirical correlations for liquefaction potential provided rather controversial results; see, for example, [12] and more recently [13].

In these calculations, thorough comparison has been made for selection of the most appropriate method for analysis. Nine of the newest and most commonly used cyclic stress based empirical correlations and two promising energy-related methods were considered initially; see Table 3.

Shear wave velocity based and, especially, energy-related methods have seen little use in practice; there is not too much experience about their application. For this reason, our focus was narrowed to the traditionally used CPT and SPT based stress methods.

The final goal of our investigation was to obtain seismically induced settlement map for the area of critical buildings and underground structures. Since the vicinity of the reactor building was explored mainly by CPTs and also because of the described above advantages of the CPT test, mapping was carried out using the CPT based methods. Since most of the empirical methods for settlement calculations rely on SPT blow counts, at those locations where CPT and SPT were performed in close proximity to each other, comparison of the settlement values obtained by the SPT and CPT based methods had been performed. One of them (signed by B3), located near the reactor building, was chosen in this paper for illustrative purposes. Soil conditions accounted for in the calculations are shown in Figure 2. The groundwater level was assumed to be 8 m below the surface.

The SPT and CPT based liquefaction assessment methods calculate factor of safety against liquefaction, that is, the ratio of cyclic resistance (CRR) to seismic demand, namely, to the cyclic stress ratio (CSR).

There are two approaches for calculating the CSR: it can be determined with site response analysis and it can be approximated by simplified equations provided for each liquefaction evaluation method. Site response analysis has to be performed by nonlinear total stress method without taking into account pore pressure increase. In many cases, nonlinear behaviour is modelled by equivalent linear method, which takes into account the degradation of the shear modulus versus shear strain by an iterative procedure. Although the real nonlinear time history analysis can be regarded as more accurate, the applicability of equivalent linear method has been studied. Its reason is that the dynamic behaviour of
Table 3: Liquefaction potential evaluation methods compared in the analysis.

<table>
<thead>
<tr>
<th>Method</th>
<th>Intensity measure</th>
<th>Empirical basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Youd and Idriss (2001) [17]</td>
<td></td>
<td>SPT</td>
</tr>
<tr>
<td>Robertson and Wride (1998) [26]</td>
<td></td>
<td>CPT</td>
</tr>
<tr>
<td>Andrus and Stokoe (2000) [28]</td>
<td></td>
<td>$V_s$</td>
</tr>
<tr>
<td>Cetin et al. (2004) [15]</td>
<td>Peak ground acceleration and magnitude</td>
<td>SPT</td>
</tr>
<tr>
<td>Moss et al. (2006) [27]</td>
<td></td>
<td>CPT</td>
</tr>
<tr>
<td>Kayen et al. (2013) [29]</td>
<td></td>
<td>$V_s$</td>
</tr>
<tr>
<td>Juang et al. (2006) [32]</td>
<td></td>
<td>CPT</td>
</tr>
<tr>
<td>Kayen and Mitchell (1997) [33]</td>
<td>Arias intensity</td>
<td>SPT</td>
</tr>
<tr>
<td>Kramer and Mitchell (2006) [34]</td>
<td>CAV$_{05}$</td>
<td>SPT</td>
</tr>
</tbody>
</table>

The building complex can be studied also using equivalent linear method because of the large complex model.

Peak ground acceleration (PGA or $a_{\text{max}}$) was computed for the design basis earthquake with $10^{-4}$/a annual frequency. Nonlinear computations have resulted in $a_{\text{max}} = 0.25$ g, while equivalent linear methods have given 0.29 g for the mean surface peak ground acceleration. From the point of view of liquefaction hazard, the moment magnitude of controlling earthquake has been determined to be equal to 6.0 using method of Marrone et al. [14]. In the simplified equations for CSR, the surface acceleration given by nonlinear method was applied.

The stress reduction factors obtained using the equivalent linear and nonlinear approaches are shown in Figure 3(a) in comparison with the results of simplified equations of Cetin [15, 16], Youd and Idriss [17], and Idriss [18]. This shows that difference between the different approaches can be significant, which can strongly influence both factor of safety against liquefaction and the resulting settlement. For this reason, it is highly recommended to use site response analysis for the evaluation of stress reduction with depth for high-risk facilities. From the simplified equations, the formula of Cetin [16] provided the best estimation of the actual behaviour of the soil column. It can be noted that some difference
can be observed between the result of equivalent linear and nonlinear approach; stress reduction factor computed by equivalent linear method decreases faster. However, in their corresponding cyclic stress ratio (CSR) values, negligible difference can be observed in the depth of interest (Figure 3(b)), because difference in stress reduction factor and PGA compensate each other. Therefore, the CSR computed by equivalent linear method has been used hereinafter to be consistent with the selected method for the soil-structure interaction analysis.

The analysis of in situ test records and resistance against liquefaction show that subsoil conditions around the reactor buildings are slightly differing from those at the control site. Around the reactors from liquefaction-induced settlement point of view, the most vulnerable layer is located mostly between 10 and 16 m in depths, but in some locations settlement was predicted in 18–20 m depths also. This finding slightly differs from the earlier results of the authors [12], where the most susceptible layer for liquefaction was computed to be between 16 and 22 m in depths on the control site. This critical depth deserves attention, because of two main reasons.

(i) The depth of 15–20 m is around the limit for which simplified procedures have been verified and uncertainty in the results can be significant.

(ii) It should be mentioned that relatively large depth of the critical layers unfavourably influences pore pressure dissipation, but on the other hand the layers are underlain by gravelly deposit, which facilitate the dissipation of excess pore pressure.

Even at those locations where CPT and SPT tests were performed in a close proximity and it is reasonable to assume that they represent the same soil conditions, factors of safety based on the two tests are rather differing (Figure 4(a)). Moreover, significant variation can be observed in the factors of safety provided by methods using the same in situ index record. High uncertainty in the SPT based methods is mainly the result of difference in the methods’ CRR-normalized blow-count correlations, which can be traced back to misinterpretation of few field cases during their development [19]. CRR-normalized CPT tip resistance correlations agree quite well with each other for relatively low seismic loading conditions, so uncertainty of the CPT based methods largely arises from the tip resistance normalization, especially from the fines content correction. Effect of two available fines content corrections on factor of safety is illustrated by an example in Figure 4(b).

After thorough consideration, the method of Boulanger and Idriss [20] was selected from the CPT based methods for mapping liquefaction potential of the reactor area. Our choice has fallen on this method because of the following reasons.

(i) Besides being deterministic, it also allows probabilistic approach of the problem, and probability of liquefaction occurrence can be incorporated into the performance-based framework of building safety evaluation.
(ii) The authors have developed CPT and SPT based procedures too and the results given by their SPT based method are the closest to the results of CPT based methods.

(iii) The probabilistic method of Moss et al. was regressed from a liquefaction case history database, but for this reason its applicability in deeper depths than 12 m is uncertain. Boulanger and Idriss have used critical state soil mechanics to extend their formula in deeper depths, which is regarded as improvement in liquefaction potential evaluation [20].

(iv) The authors of the method have been continuously revising and updating their method with their latest results for approximately 10 years. They published revised corrections and updates in 2008, 2010, and 2012 and recently in 2014 including liquefaction case histories from the most recent earthquakes of 2010-2011 Canterbury earthquakes and 2011 Great Tohoku earthquake. Thus this method can be considered as the most up-to-date correlation.

(v) As the task was not a design problem but was the assessment of an existing building, at the selection of methods, we aspired to limit the conservatism involved in the calculation.

3.3. Settlement due to Liquefaction. The main goal of the evaluation is to determine the anticipated postliquefaction displacements. As most of the empirical settlement calculation methods are based on SPT blow-count numbers, mainly these methods had been used in earlier studies. Because many CPT tests were carried out around the main building complex of Paks NPP and CPT has many advantages over SPT, CPT based liquefaction potential and settlement evaluation method was used finally.

In the frame of preparatory studies for Paks NPP site, several methods for settlement computation have been compared: SPT based methods of Ishihara and Yoshimine [21], Tokimatsu and Seed [22], Wu and Seed [23], and Cetin et al. [24], as well as the CPT based method of Zhang et al. [25]. Most of these methods rely on factor of safety against liquefaction and/or normalized penetration resistance, so they have to be used in conjunction with liquefaction potential evaluation methods. The first two methods are compatible with the methods of Youd et al. and Idriss and Boulanger methods, while the Wu and Seed procedure can be used with the method of Cetin et al.

Cetin et al. [24] used a new approach to develop their correlation for empirical settlement analysis. Instead of using laboratory results, high-quality cyclically induced ground settlement case history formed the base of their method, which allowed probabilistic assessment of the database. Their
procedure is based on CSR and SPT blow-counts normalized to energy, overburden pressure, and clean sand. The method proposes the use of a depth weighting factor, which takes into account the observation that deeper layers play less important role in the surface settlement. Their statistical assessment showed that the optimum value of this threshold depth is 18 m. Because of these features, this method can be considered as most appropriate among the investigated ones.

From the CPT based settlement calculation methods, only one option, the procedure of Zhang et al. [25], was available for the analysis. This method computes the volumetric strain from factor of safety against liquefaction and CPT tip resistance normalized to clean sand. The authors proposed the use of Robertson and Wride [26] method to compute the factor of safety, which was at that time the state-of-the-art CPT method.

According to Cetin et al. [24], we have limited the depth of settlement calculations but the threshold depth was taken more conservatively to 20 m. For B3 point, the procedure of Zhang in conjunction with the methods of Robertson and Wride [26], Moss et al. [27], and Boulanger and Idriss [20] resulted in the following settlements: 0.8 cm, 9.2 cm, and 1.6 cm, respectively. We used the method of Boulanger and Idriss for mapping because of the reasons presented above in Section 3.2.

Comparison of these values with the results of SPT based calculations showed that in general all of the SPT based settlements were significantly larger than the CPT based values. The largest settlement was predicted by Ishihara and Yoshimine as well as by the Tokimatsu and Seed methods, while Cetin et al. gave the lowest values. After thorough revision of the results, methods, and tests, it was noted that probably some kind of error distorts the SPT records in a few test points. In those places where all CPT, SPT, and Vₚ records showed very similar sequence of stiffer and softer layers, but SPT blow-count numbers have contradicted to that stratigraphy.

The extent of the geotechnical survey allowed the assessment of lateral variability of soil conditions. Mapping of seismically induced settlement was based on altogether 29 CPT records around the reactor buildings and the values were varying between 0.1 cm and 5.1 cm. However, this maximum value is quite outlier, because in most of the test points less than 1 cm settlement was predicted, and the settlement exceeded 2 cm in only four test points.

The free surface settlement was computed using effective stress method [19] to the average soil profile. It gave 0.67 cm as average settlement, which is consistent with the results given by the combination of Zhang et al. and Boulanger and Idriss methods.

4. Conclusion

In the paper, an important application of liquefaction assessment is discussed: beyond design base analysis of liquefaction consequences for nuclear power plants. Detailed framework for performing the safety analysis for liquefaction consequences is outlined in the paper. Deterministic safety analysis of nuclear power plant for earthquake-induced liquefaction is a complex task that requires adequate modelling of the plant response and characterization of the hazard and engineering demand parameter of the liquefaction, as well as the assessment of the integrity and function of plants systems, structures, and components. Preparatory analyses and considerations show that the settlement could be the dominating engineering demand parameter for the case of Paks NPP site. Adequacy of safety analyses and conclusiveness of the results is mainly limited by the epistemic uncertainty of the methods of hazard definition and the engineering parameters characterising the consequences of liquefaction and controlling the plant response. In the paper, detailed comparison of available methodologies has been made for adequate selection of methods for calculation of the settlement.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


Research Article

Probabilistic Dynamics for Integrated Analysis of Accident Sequences considering Uncertain Events

Robertas Alzbutas

Lietuvos Energetikos Institutas, Breslaujos 3, LT-44403 Kaunas, Lithuania

Correspondence should be addressed to Robertas Alzbutas; robertas.alzbutas@lei.lt

Received 16 January 2015; Revised 14 April 2015; Accepted 16 April 2015

Academic Editor: Francesco Di Maio

Copyright © 2015 Robertas Alzbutas. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The analytical/deterministic modelling and simulation/probabilistic methods are used separately as a rule in order to analyse the physical processes and random or uncertain events. However, in the currently used probabilistic safety assessment this is an issue. The lack of treatment of dynamic interactions between the physical processes on one hand and random events on the other hand causes the limited assessment. In general, there are a lot of mathematical modelling theories, which can be used separately or integrated in order to extend possibilities of modelling and analysis. The Theory of Probabilistic Dynamics (TPD) and its augmented version based on the concept of stimulus and delay are introduced for the dynamic reliability modelling and the simulation of accidents in hybrid (continuous-discrete) systems considering uncertain events. An approach of non-Markovian simulation and uncertainty analysis is discussed in order to adapt the Stimulus-Driven TPD for practical applications. The developed approach and related methods are used as a basis for a test case simulation in view of various methods applications for severe accident scenarios and uncertainty analysis. For this and for wider analysis of accident sequences the initial test case specification is then extended and discussed. Finally, it is concluded that enhancing the modelling of stimulated dynamics with uncertainty and sensitivity analysis allows the detailed simulation of complex system characteristics and representation of their uncertainty. The developed approach of accident modelling and analysis can be efficiently used to estimate the reliability of hybrid systems and at the same time to analyze and possibly decrease the uncertainty of this estimate.

1. Introduction

A number of different methodologies were proposed in order to analyze stochastic events and the time intervals that elapse between them. The most known theoretical background of these methodologies to treat and analyze dynamic systems was still based on the Markov approach. For instance, the Theory of Probabilistic Dynamics [1–3] was extensively investigated in order to perform analytical modelling and simulation related to the analysis of system reliability and safety. Dynamic reliability techniques [4] have been developed in order to study the reliability parameters of complex dynamic systems having continuous processes and discrete events (e.g., failures) interacting with each other.

In dynamic reliability theory, the concept of reliability includes the interaction existing between the sequence of dynamics and events, such as the crossing of the border of a safety domain in the space of the physical variables and the transitions between dynamics. The large number of states, possible time-dependent delays, and transition probabilities that are to be evaluated may be the most important limitation of using the Markov approach for large sets of system components. In addition, the stochastic events and uncertainty in the parameters of the dynamics complicate the analysis even more. Thus, uncertainty analysis becomes necessary and some general uncertainty estimation and analysis techniques have been further introduced and discussed.

The large parts of commonly used methods for reliability analysis and probabilistic safety assessment (PSA) are usually based on the assumption that the basic events are functionally independent of each other. This assumption does not often hold, and Markov processes are mainly used to account
for the time dependence of the reliability and availability functions.

In this case, it is possible to use an assumption that the transfers from state to state follow a Markov process. The initial equations to be considered may be expressed as follows:

\[
\frac{d\pi_j(t)}{dt} = -\lambda_j\pi_j(t) + \phi_j(t),
\]

\[
\lambda_j = \sum_{k \neq j} p_{j \rightarrow k},
\]

\[
\phi_j(t) = \sum_{k \neq j} p_{k \rightarrow j}\pi_k(t),
\]

where \(\pi_j(t)\) is the probability of the system being in state \(j\) at time \(t\) and \(j\) is the system state vector, composed of the set of system states. In (2), \(\lambda_j\) as the total transition rate out of state \(j\) is defined by sum of all \(p_{j \rightarrow k}\), which are the transition rates from state \(j\) to state \(k\). The term \(\phi_j(t)\), as defined in (3), is called the ingoing density, that is, the instantaneous frequency at which state \(j\) is entered from any other state at time \(t\).

In general, each state can be associated with specific evolution equations for the process variables describing the system dynamics. But, in a Markov process, the probability for the system to stay in a given state during a given sojourn time is independent of the time at which the state is entered, so state probabilities are independent of the past history (memoryless stochastic systems).

The assumption of a Markov approach (i.e., the assumption that the probability that a system will transfer from one particular state to another depends only on the initial and final states of the transition) holds major simplification of the simultaneous equations describing the state space diagrams.

However, even these simpler equations may not be soluble in analytical form, if the transition rates and possible delays between states are time-dependent functions. The analysis is even more complex, if transition rates are uncertain, that is, depend on uncertain parameters.

Initially, for uncertainty analysis and simulation of complex processes, the aggregate approach and the method of control sequences have been investigated and widely used. According to this approach, the simulation of dynamic systems and integration of modelling methods were also considered [5]. According to this approach, the investigated objects are presented as the set of interacting Piecewise Linear Aggregates (PLAs) [6]. The method of control sequences is used for the aggregate specification. Initially, PLA formalism was mainly used for discrete event system specification and analysis of distributed systems [7]. Later on, applying the advantages of PLA, the focus was set on simulation and analysis of hybrid systems considering the stimulated dynamics and interactions with various events [8]. However, the practical application of this approach showed various limitations and a need to look for different approaches or integration of them.

Due to the strong dependence existing along an accident scenario between stochastic events (e.g., operator actions or component failures) and dynamics (i.e., the time-dependent evolution of physical processes, e.g., a change in temperature during a transient), the traditional simulation using discrete PLA formalism or Markov processes is not able to cope with such time-dependent hybrid system simulation. The reliability analysis of the system is even more complex, if transition rates are uncertain, that is, depend on uncertain parameters. Thus, extended approaches are considered in order to cope with this issue and the uncertainty analysis.

The paper, based on a short article in proceedings of conference [9], is constructed as an investigation of issues in probabilistic dynamics to give for the reliability analyst and PSA practitioner a wider and clearer view of how accident sequence analysis considering uncertain events can be performed. This is very relevant for level-1 PSA [10] and especially for level-2 PSA [11]. It is worthwhile to mention that there are various techniques of dynamic event tree generation (e.g., ADAPT [12], MCDET [13]), which are specifically useful for level-2 PSA. The main contribution of this paper to the reliability assessment field is in wider discussion of stimulus-driven treatment of probabilistic dynamics and in practical application of related methods and development of approach for severe accident scenario simulation and uncertainty analysis, which was demonstrated by the test case and extension of its initial specification. This can be further used in the benchmark exercise for comparison of other methods and approaches.

The structure of paper is as follows: after the presentation of introduction and this outline of the paper as well as considered methods and issues (Sections 2.1 and 2.2) the formal concept of stimulated dynamics and dynamic systems is presented in Section 2.3; further, in Section 2.4 the modelling and simulation approach as well as analysis tool is introduced being more specific on implementation algorithm and stimulated dynamics treatment. The proposed approach for analysis of uncertainty issues is emphasized in Sections 3.1–3.3, where the uncertainty estimation and analysis taking into account sensitivity measures as well as the concept for implementation of integrated analysis using coupled software are presented. A practical part of the paper (starting from Section 4.1) is devoted for the test case analysis. Initially, the test case specification is focused on process timing and associated events with its relation to the stimulated dynamics concept accordingly presented in Section 2.1. Then, this case study (in Section 4.2) is presented and for comparison purpose (in Section 4.3) is related to the event tree and simplified analytical modelling. The simulation of time-dependent rupture, which was the main concern in the test case, is described in Section 4.4. Finally, the analysis of uncertain rupture frequency is presented in Section 4.5. In the same section the results of time-dependent uncertainty and sensitivity analysis are demonstrated and related to the idea of how this can be used in order to focus on the rarest sequence with the quite severe consequences and possibly reduce the computational time of simulations performed. Then, by discussions and conclusions, the paper is completed.
summarizing the related PSA issues and advantages of proposed approach application for accident sequence analysis considering hybrid systems and uncertain events.

2. Stimulated Dynamics and Uncertain Dynamic Systems

2.1. Extensions of the Markov Approach. Extensions of Markov processes initially have been developed for cases where the transition rates depend on process variables, that is, when the TPD is valid [14]. Indeed, in some unfortunately frequent cases, there is a stochastic time delay between events (e.g., satisfaction of ignition conditions and explosion itself). For instance, operators introduce delays in taking actions after alarms that may lead to different further accident developments. In these cases, stochastic delays complicate the situation of the state transitions.

More generally, the same situation occurs whenever, for a transition (event) to occur, some conditions ought first to be fulfilled that depend on the accident transients and timing. These conditions may persist after the transitions to the new states.

A typical example is the occurrence of combustion phenomena only if flammability conditions are met, with delays potentially resulting from stochastic ignition conditions and with potential for multiple combustions if the flammability conditions persist. For instance, this can be actually related to the hydrogen generation and possibility of combustion during the severe accident [15]. The more general conditions (including setpoints for thresholds as particular cases) may be considered as stimuli for the transitions (events). When process variables reach those conditions, the stimulus activation or start of the delay before the transition can be considered.

Because stimulus activation conditions the occurrence of events, the history of stimulus activations and subsequent delays during the event sequence does matter in calculating the scenario frequencies; extensions of the Markov process equations accounting for these features are then necessary. Those extensions constitute the so-called Stimulus-Driven Theory of Probabilistic Dynamics (SDTPD).

Indeed, SDTPD [14] provides a mathematical basis to estimate the probabilities per unit time of entering states with specified activated stimuli and subsequent delays. Exceeding safety objectives is a particular case of stimuli, so SDTPD considering various stimuli has the potential for analyzing multiple objectives, including safety [16].

More recently, the simplification of SDTPD or Theory of Stimulated Dynamics (TSD) was developed for the analytical modelling and the simulation of hybrid (continuous-discrete) systems [8, 17]. The theory at first deals with instantaneous and random variations of process variables; then, it introduces the concept of stimulus and how it can be implemented [14]. Both a semi-Markov and a non-Markovian treatment may be used in order to adapt TPD for practical applications, mostly in the context of PSA. The development of TSD as well as related methods and simulation methodologies has been used by the TSD developers as a basis in the perspective of their applications for PSA and severe accident analysis.

Since the application of TSD-related methods to the traditional PSA concept [18] needs a formal approach, the new definitions and issues of uncertainty analysis are specified and discussed. Then the related investigation of reliability and uncertainty analysis is performed.

2.2. Issues of Uncertainty Analysis. The part of uncertainty related to any estimation can be considered as a spread or distribution in the value of the result estimate. Obviously, the spread in this estimate is related to the spread in the parameters of the probabilistic model used to estimate the result, for example, risk [16].

However, in addition to uncertain model parameters, another cause of uncertainty may arise from incompleteness, that is, from the incomplete modelling or data used in the probabilistic model itself or in the analyses used to derive the model. The uncertainty in inputs may also affect the topology of the probabilistic model or the data and time dependence used to quantify it.

The completeness of the scenario inventory depends on the consideration of each scenario construction, that is, on the way of grouping sequences and assigning to them corresponding frequencies and consequences. The level of conditional risk, given the scenario occurrence, partly represents how rare and important this scenario is from a consequence point of view. However, in order to search for scenarios with almost unpredictable, but possibly severe, consequences, there is a possibility to generate and consider events and dynamics as well as related sequences, which can be very rare.

The technique to search for rare random events is not evident and is not related to traditional PSA; in addition, there is also a concern about how to generate and to consider rare events, which are dependent on the changes in process variables values and timing. Actually, this means that scenarios related to such events are time-dependent and uncertain. Thus, for uncertainty analysis, the scenario development should be considered as well as uncertain parameters.

Uncertainty related to PSA could be classified according to the uncertainty source: the frequency of events and the sequences of dynamics themselves (i.e., dynamics and timing in the process variables space). Taking into account that all sources of uncertainty are important, there is a need for such uncertainty analysis, which considers both sources and at the same time reflects the issue of model incompleteness.

On the basis of this classification, there is a need to note that in the case of the first source of uncertainty, changing the values related to the frequency of event occurrence (e.g., failure rate \( \lambda \)) will not create new branching situations. It will affect the likelihood of already possible sequences and scenarios [19], without changes in the possible process variables evolutions and scenarios themselves. Conversely, a change in the value of an uncertain threshold related to a specific event creates a new dynamic trajectory in the process variables space.

Considering this classification, it is easy to conclude that the first source of uncertainty (i.e., fluctuations of failure or recovery rates or of on-demand failure probabilities) can be propagated with no additional deterministic calculations, as all sequences in the process variables space keep valid.
However, the second source of uncertainty, in principle, causes a continuum of additional scenarios with different timing, what requires considering probabilistic dynamics [4]. This uncertainty has an effect on the process variables timing, what requires considering probabilistic dynamics causes a continuum of additional scenarios with different evolutions and it should therefore be investigated separately in order to save computational resources and represent conditions and scenario-related uncertainty or simulation incompleteness.

Taking into account the main features of dynamic systems, it can be seen that a simulation algorithm relevant for dynamic reliability and uncertainty issues should display the following characteristics:

(i) Search for rare conditions under consideration.

(ii) Representation of the uncertainty of the conditions considered.

2.3. Formal Concept of Stimulated Dynamics. In the considered case, the modelling and analysis of dynamic systems is related to stimulated dynamics. Dynamics is determined by laws of process variables evolution, which can be indexed by an integer \( i \in N \). Process variables \( \bar{x} \) can be governed by a set of deterministic equations; that is, \( d\bar{x}/dt = g_i(t, \bar{x}(t)) \), \( \bar{x}(0) = \bar{x}_o \), and \( \bar{x} \in R^N \). In general, \( g_i \) is the dynamic model pertaining to the \( i \)th configuration and driven by the vector of physical variables \( \bar{x} \).

An instantaneous change of the dynamics due to stimulus activation and subsequent delay elapsing is associated with an event. Event \( e \) is defined as a transition between dynamics at a certain time \( t \). A random event is an event whose occurrence is related to complex nature and timing, which is modelled stochastically, for example, a time distributed failure occurrence. A deterministic event is induced by deterministic rules (analytical equations).

To relate event with stimulus, there is a need to explain that a stimulus covers any situation or conditions whose occurrence, after a time delay, potentially causes an event to occur. An example of such an event can be related to the time moment when following a given process a threshold on pressure \( p \) is reached and safety functions after the delay (e.g., operator reaction) are activated (see Figure 1).

In the usual formulations of the Theory of Probabilistic Dynamics, the change in the dynamic behavior of the physical process variables occurs with no delay. The main concept introduced here is the stochastically determined succession of stimulus activation and delay, which must take place prior to the transition between two dynamics, that is, the actual occurrence of an event.

Formally, for analysis, let \( \Phi \) be the set of all stimuli to be accounted for in the process evolution following the occurrence of a given event related to the transitions between dynamics. Denote by \( f_i^F(t_F; \bar{\omega}) \) the probability density function of activating stimulus \( F \in \Phi \) at states \( \bar{\omega} = g_i(t_F, \bar{\omega}) \) after time \( t_F \) spent in dynamics \( i \) which was entered at state \( \bar{\omega} \) (see Figure 2 [14]).

Also define \( h_{ik}^F(t_{dk}; \bar{\omega}) \), probability per unit time of having a time delay \( t_{dk} = t_F - t_{F_k} \) between the activation of stimulus \( F_k \) at time moment \( t_{F_k} \) and the actual occurrence of event at time moment \( t_F \), that is, transition of dynamics \( i \rightarrow k \), if stimulus \( F \) was activated at state \( \bar{\omega} = g_i(t_F, \bar{\omega}) \):

\[
h_{ik}^F(t_{dk}; \bar{\omega}) = \sum_{k \neq i} h_{ik}^F(t_{dk}; \bar{\omega}),
\]

where \( h_{ik}^F(t_{dk}; \bar{\omega}) \), probability density function of the delay \( t_{dk} \) between the activation of stimulus \( F \) at time moment \( t_F \) and in the same conditions leaving dynamics \( i \) at time moment \( t_{F_k} \).

2.4. Simulation Approach and Analysis Tool. After the occurrence of an initiating event, the corresponding evolution laws of the process variables are considered. They induce stimuli activations with their corresponding delays and related events. Such a process is carried on until the considered process variables reach one state among possible final absorbing end states. In reliability analysis, an end state is a consequence expressed as a damage state or as a stable safe situation. The probability estimate (frequency) of any consequence practically can be calculated as it develops during the simulation of the considered system.

The SDTPD is a good candidate for level-2 PSA because it mitigates the weaknesses of the current methods. But the direct application of this theory requires solving a lot of complex equations. Even in the simple test case presented below, it is not possible to solve them analytically. Moreover, for large systems, it is unthinkable to want to write the associated equations.

Analyzing complex dynamic systems and accidents, analytical methods often cannot be properly applied. In such cases, Monte Carlo simulation can be used. It is based on random numbers generation. The accident probability estimation is determined according to the rate of success/failures and the number of trials. The most important part of this approach is to develop the simulation model of the considered physical system and stochastic process.

For instance, considering a simple dynamic system, which includes a stimuli \( F \) and one possible event changing the dynamics, there is a need to simulate the deterministic process (changes of variables) \( \bar{x} = g_i(t, \bar{\omega}); \bar{\omega} = \bar{x}_{i_{t=0}} \), and the occurrence of the event related to the stimulus \( F \)’s activation...
and the subsequent time delay. Such system simulation can be expressed using the following algorithm:

(i) Process variables follow the dynamics law \( \mathbf{x} = \mathbf{g}(t, \mathbf{x}, \mathbf{w}), t > t_0 \).

(ii) On the (possibly random) time moment \( \tau_F \) the stimulus \( F \) is activated.

(iii) During the random time delay \( \tau \) following the stimulus activation, the process variables still follow the same dynamics.

(iv) On time moment \( t_F = \tau_F + \tau \), the event occurs and the process is changed to another.

This is a really simple algorithm, which is based on two random variables: the time moment \( \tau_F \) of stimulus \( F \)’s activation and the time delay \( \tau \). In order to generate the values of these random variables one must know the probabilistic distribution functions \( f^F(t) \) and \( h^F(t) \). Considering complex systems, similar functions usually are not known as they depend not only on time but on the process variables value as well.

Assuming that the following distribution functions of stimulus activation \( f^F(t, \mathbf{x}) \) and time delay \( h^F(t, \mathbf{x}) \) are known for each dynamics \( i \) separately, it must be noted that these functions depend on process variables \( \mathbf{x} \) and this does not allow us to determine, in advance, when stimulus \( F \) will be activated and how long the delay will take. The time moment of stimulus \( F \)’s activation \( \tau_F \) is generated as follows:

1. A pseudorandom number \( p \) is generated according to the uniform distribution from the interval \([0; 1]\).
2. According to the considered dynamics \( i \) and related process variables \( \mathbf{x} \), the corresponding distribution function \( f^F_i(t, \mathbf{x}) \) is determined for any time moment \( t \).
3. The time moment of stimulus \( F \) activation \( \tau_F \) is determined from equation \( p = \int_0^{\tau_F} f^F_i(t, \mathbf{x})dt \).

In order to estimate process variables \( \mathbf{x} \) and distribution function \( f^F_i(t, \mathbf{x}) \) of stimulus \( F \) activation for any time moment, many computations are needed. Thus, in order to save resources, the calculations are made using discrete time framework. \( \tau_F \) is determined by performing the following actions:

1. A pseudorandom number \( p \) is generated according to the uniform distribution from the interval \([0; 1]\).
2. The time related probability \( s(t) \) initially is defined as follows: \( s(t_0) = 0 \).
3. The discrete time \( t \) is considered starting from \( t_0 \) step by step \( \Delta t : t_i = t_{i-1} + \Delta t \).
4. During each time step \( t_i \) the following computations are performed.
   (i) The process variables \( \mathbf{x} \) are estimated.
   (ii) According to the considered dynamics \( i \) and values of process variables \( \mathbf{x} \), the probability \( p_i \) that stimulus \( F \) will be activated or deactivated during the time interval \([t_i - 1; t_i]\) is computed according to SDTPD.
   (iii) Consider \( s(t_i) = s(t_{i-1}) + p_i \).
   (iv) If \( s(t_i) \geq p \), then time moment of stimulus \( F \) activation \( \tau_F = t_i \).

The described algorithm allows generating the time moments without direct (analytical) consideration of changes in process variables and distribution function \( f^F(t, \mathbf{x}) \). This algorithm can also be used for the generation of the random time delay \( \tau \) after stimulus activation.

Based on this approach and in relation to the theory of stimulated dynamics, a simulation tool was developed. In this tool, the building of the system model with stimulated dynamics is based on three groups of essential characteristics:

(i) Timing: it is the independent factor, which increases gradually and is related to every event. Time influences values of process variables, stimulus activations, delays, system events, and so forth.

(ii) Deterministic characteristics: process variables and dynamics, which indicates the system state at a certain time moment.

(iii) Stochastic characteristics: probabilistic density functions of stimulus activation and delays before events.

(iv) The relationships between these characteristics are shown in the following schematic (see Figure 3).

With reference to this schematic, three different modules are distinguished in the simulation tool:

(i) Timing and events’ control module.

(ii) Deterministic process module.

(iii) Stochastic process module.

In general, the timing control module simulates the system timing. The deterministic processes module estimates the process variables evolution and according to the considered dynamics defines the system state. The stochastic processes module estimates the probabilities of stimulus activations and of dependent delays defining the following events.

TPD enables analyzing systems, where transition between different dynamics is initiated immediately, when some critical conditions (reaching of threshold) occur. In addition, the application of TSD enables analyzing such systems, where delays before events are stochastically determined as in real systems with uncertain time between the stimulus activation and the start of a new dynamics.
3. Uncertainty Analysis Approach

The approach suggested for uncertainty and sensitivity analysis is based on well-established concepts and tools of probability calculus and statistics. In this paper, it is illustrated by an application of SUSA (Software System for Uncertainty and Sensitivity Analyses) developed by GRS [20]. The uncertainty analysis, in addition to uncertainty estimation, includes the identification of the potentially important contributors to the uncertainty of the model output and the quantification of the respective state of knowledge by subjective probability distributions [21].

In general, the probabilistic model expresses aleatory (stochastic) uncertainties of physical process. In addition, for each uncertain input of the model, its probability distribution also expresses how well input is known (i.e., epistemic uncertainty). The sensitivity analysis, as main part of the uncertainty analysis, can be used to identify and classify uncertain parameters, which mainly contribute to the variations of results and in order to see the uncertain input’s combined influence on the output due to uncertainty propagation. For this, the quantification of margins and uncertainties is essential [12].

3.1. Uncertainty Estimation and Analysis. The quantitative uncertainty estimation can be expressed using quantiles or percentiles (e.g., 5% and 95%) of the probability distribution. Knowing distribution law and parameters, it is possible to estimate the mean, standard deviation, median, quantiles, and other point estimates as well as confidence intervals. In practice, quantiles of output can be estimated using Monte Carlo simulations (MCS) with a specified number of model runs after input sampling.

In addition, the impact of possible sampling error on the output can be considered and related to the number of runs. This can be done by computing \((\alpha, \beta)\) statistical tolerance limit (or two sided limits treated as interval). This limit (or interval) separates at least \(100 \cdot \alpha\%\) part of all possible output with at least a \(\beta\) probability as confidence level. In other words, this means that, with a \(\beta\) probability, \(100 \cdot \alpha\%\) part of all possible output will be separated by the specified statistical tolerance limit (or will be in the considered statistical tolerance interval).

According to the classical statistical approach, the confidence statement expresses the possible influence of the fact that only a limited number of model runs have been performed. For example, according to Wilks formula [21], 93 runs are sufficient to have a \((0.95, 0.95)\) statistical tolerance interval (upper and lower limits). The required number \(n\) of runs for one sided \((\alpha, \beta)\) tolerance limit and correspondingly the number \(n_2\) for \((\alpha, \beta)\) tolerance interval can be expressed as follows:

\[
\begin{align*}
  n_1 & \geq \frac{\ln (1 - \beta)}{\ln (\alpha)}, \\
  n_2 & \geq \frac{(\ln (1 - \beta) - \ln ((n_2/\alpha) + 1 - n_2))}{\ln (\alpha)}.
\end{align*}
\]  

The minimum number of model runs needed is independent of the number of uncertain quantities taken into account and depends only on the two quantities \(\alpha\) and \(\beta\) described above.

3.2. Uncertain Output Sensitivity Analysis. In general, outputs from models are subject to uncertainty. Usually uncertainty estimation can provide a statement about the separated or combined influence of potentially important uncertainty (aleatory and epistemic) sources on the model output. However, often more important, to analyze uncertainty providing quantitative sensitivity statements that rank the uncertain inputs with respect to their contribution to the model output uncertainty. On other hand, it is important to note that uncertainty in the model affects the ranking results [22]. In a frame of uncertainty analysis the purpose of the considered sensitivity analysis is

(i) to analyze uncertain output sensitivity to the uncertain inputs,

(ii) to identify which inputs mostly influence the model output.

In general, sensitivity analysis is used not only to analyze uncertainty but also to examine which epistemic uncertainty sources are better to control.

In order to rank uncertain parameters according to their contribution to model output uncertainty, standardized regression coefficients (SRCs) [23] can be chosen from the many other sensitivity measures available. They are capable of indicating the direction of the contribution (negative means inverse proportion). SRC is supposed to tell by how many standard deviations the model result will change if the uncertain input is changed by one standard deviation.

Additionally, the correlation ratios (CRs) [23] can be computed. The ordinary CR is the square root of the quotient of the variance of the conditional mean value of the model output (conditioned on the uncertain input) divided by the total variance of the model output due to all uncertain input taken into account. It serves as a measure of how one uncertain model specification was quantified through a set of alternative specifications. The CR quantifies degrees of inputs and output relationship.

How well this is achieved in practice depends on the degree of linearity between the model output and the uncertain input. In case the number of uncertainties is large and the sample size is small, spurious correlations can play a nonnegligible role. The effect of spurious correlations on sensitivity measures may be investigated if the estimates of SRCs and correlation coefficients are compared [21]. Thus, often in practical cases SRCs are also applied with other sensitivity measures, like Partial Correlation Coefficient (PCC) [24]. Correlation may provide a measure of the strength of a linear association between variables and results. For nonlinear but monotonic relationships between results and variables, measures that work well are based on rank transforms. The PPC provides just a measure of variable importance that tends to exclude the effects of other variables [23]. The options described above and chosen for this illustration of the
3.3. Integrated Analysis Using Coupled Software. In order to perform the uncertainty analysis of a probabilistic model output, the double randomization MCS scheme can be applied. The dynamics simulation is proposed to be performed, using different values of the random input defined in Uncertainty Simulation Software denoted as USS. The considered Dynamics Simulation System, here denoted as DSS, is used for the dynamics simulation in relation to probabilistic system failures, accidents, and/or consequences. The used USS interacts with DSS and performs an additional statistical analysis; that is, outputs from DSS are transferred to the USS system, which then perform an integrated uncertainty and sensitivity analysis.

The major part of the integrated analysis is based on the coupling translator, which consists of a preprocessor and a postprocessor used for data flow between dynamics and the uncertainty simulation software. In general, four main steps are used to perform an integrated analysis:

1. Develop a probabilistic model (e.g., TSD based).
2. Sample model inputs (e.g., uncertain delays).
3. Simulate the stochastic process (e.g., pressure).
4. Analyze outputs (e.g., rupture frequency).

The integration starts, when DSS input file, which represents the model, is processed by a translator, using a preprocessor to provide USS with information, which is available in the DSS model.

Most of the time, when the translator is running, it modifies the identified random variables in the DSS input file according to the values, determined by USS, and retrieve the values of response from the DSS output file for the following uncertainty analysis by USS. At the end of the computations, USS may perform the uncertainty analysis. The following scheme (Figure 4) graphically shows the interactions between the user, USS, DSS, and the translator.

The first step in using USS with a coupling of DSS is to set up the model uncertainties. This is done by selecting uncertain inputs, specifying distributions, and so forth. This part depends on the type of uncertainty analysis performed. In general, uncertain inputs and specific distributions with their parameters are specified. The list of considered inputs depends on quantities from DSS, which are available in input files and are proposed to be treated either as stochastic or epistemic. The distribution functions and distribution parameters for each uncertain input are specified. Then, prior to each model call (i.e., DSS activation) corresponding values from the DSS input file are modified by the translator according to the values selected by USS. In the final stage, the failure criterion and responses as considered output quantities (e.g., pressure) are analyzed in order to estimate reliability parameters (e.g., failure rate or frequency).

4. Test Case Simulation and Analysis

4.1. Test Case Specification. The extension of initial test case is specified in relation to a benchmark exercise [26] defined in the framework of the SARNET (Network of Excellence for a Sustainable Integration of European Research on Severe Accident Phenomenology) [27], dedicated to severe accident analysis and PSA methodologies development. The main task of the benchmark was to quantify the risk of containment rupture frequency (estimate of containment failure probability) due to hydrogen combustion. As a “basic transient” the following simplified object/situation was considered: A French 900 MWe Pressurized Water Reactor (PWR) with 3 loops and Passive Autocatalytic Recombiners (PAR) operating at nominal power before the initiating event; Loss of Coolant Accident (LOCA) after a 3'' break size on cold leg of Reactor Coolant System (RCS); failure of all water injection system and spray system. This issue was supposed to be most relevant for level-2 PSA.

The aim of the benchmark was to provide a simple example that demonstrates the limitations of the classical level-2 PSA methods and to assess the expected improvements that could be obtained by dynamic reliability methodologies against the classical PSA approach. This was done in order to test, validate, and compare the different methodologies mentioned above and within the framework of the SARNET [27].

The main part of benchmark [26] has been divided in 2 steps with progressive complexity in both the probabilistic
and the dynamics aspects of the sequences: Step-1: first implementation of the problem with dynamic or classical method, with simple analytical model for the physics; Step-2: second implementation of the problem with complements in the analytical models for epistemic uncertainties.

The steps 1-2 comparison report has already been issued that indicates that it is essential to know the details of the interpretation made of the benchmark specification by the respective groups. Possible further activities were also considered, like implementation of ASTEC (Accident Source Term Evaluation Code) modules instead of simple analytic model. Although two main conclusions regarding classical PSA treating the issue of various possible chronological events have been drawn from the benchmark

(1) “the treatment of all chronological issue is difficult,”

(2) “the treatment of multiple combustions is impossible.”

Consequently, the summary of this is that it is not possible to solve the benchmark with a classical event tree. The conclusion is that "the only way for a classical approach is a conservative approach.”

The study and solution of the benchmark might as well help to gain insights of the differences of the methodological approaches, its advantages, and its limitations, both for the theoretical and the practical issues involved in the development of the different approaches. By doing this, some model deficiencies could be as well detected and corrected for future considerations.

In this context, and also as part of the SARNET goals, some activities were extended, as, for instance, the so called Step-2 plus, and Step-3 was defined and additionally partly performed looking for the further benchmark.

The present test case is the result of the work done over two main objectives: to discuss and demonstrate the benchmark exercise and the details of its design and to acquire a deeper knowledge of SDTPD and Monte Carlo simulation techniques and uncertainty assessment methods and its particular application to the further benchmark, to investigate the potential of these techniques as a complement of the dynamic reliability techniques. This test case is therefore also divided in two main parts.

The first part, presented in this section, is dedicated to remember the description, specifications, and assumptions of the benchmark, as well as the solutions presented by the partner organizations which used probabilistic dynamics. In particular, it is focused on the solution developed by Lithuanian Energy Institute (LEI) and Université libre de Bruxelles (ULB) by means of SDTPD approach. Finally, some assumptions are revised and reinterpreted and specifications are extended for the further benchmark.

The other parts of this section concern the performance of the test case, mainly the Step-2 and Step-2-plus, where uncertainty and sensitivity analysis has been performed to study the influence of the uncertainties on the final result. The previous section introduces a brief description of approach used for uncertainty and sensitivity analysis.

The test case as benchmark exercise was divided into two consecutive steps, which are related to each other. Step-1 is the basic part used for basic probabilistic model development and point estimate calculations. Step-2 contains more details and extensions regarding random events and uncertain inputs; thus, this makes the modelling and a point estimate more realistic.

The single result as a point estimate can be quite precise due to a lot of histories in single simulation; however, for the practical analysis of physical behaviour and application of it for safety purposes, the investigation of variation of results itself is quite important. Such investigation is based on sensitivity and uncertainty analysis of sampled results and is a kind of a Step-2-plus assessment case. This research can show which uncertain parameters are the most important in order to change the behaviour of aleatory phenomena and decrease the uncertainty of the results.

The input parameters can also be related to the timing; that is, the set of input parameters can include uncertain delays and randomly distributed time moments of events occurrence. However, this was not a case in the Step-2 of benchmark exercise. For such investigation, the extended benchmark exercise could be used.

Further, in the next step (Step-3) or in the new benchmark, the focus could be more on epistemic uncertainties and timing parameters (e.g., random delays) when performance of additional analysis of modelling uncertainty could be a good demonstration of timing and modelling importance for the estimation of safety related result (in the similar benchmark case, the probability of containment rupture).

The practical test case used for investigation, as mentioned previously, concerns the containment failure risk assessment due to hydrogen combustion for a French 900 MWe PWR (3 loops with Passive Autocatalytic Recombiners). The specification is based on the results of two transitions (transitional processes) computed by Institut de Radioprotection et de Sûreté Nucléaire (IRSN) with ASTEC software package (Accident Source Term Evaluation Code) in which there is no water injection before core dewatering, the first one without spray system activation and the second one with spray system activation. Actually, the ASTEC calculations also provided the timing and basic information on

(i) the kinetic of core degradation process;

(ii) the kinetic of hydrogen and vapour releases in containment;

(iii) the delay before vessel rupture;

(iv) the pressure evolution in containment (and atmosphere composition).

As it was mentioned, the different phases of the test case have been separated in several steps. Step-1 concerns the basic example, changing only the failure criterion. Step-2 contains some more details and extensions regarding some stochastic events in order to express some source of uncertainties in the initial data. Step-3 could be related to the reflection of epistemic uncertainties including uncertain modelling and different timing effects.

In general, after initial Loss of Coolant Accident (LOCA), there are three safety systems taken into account: Passive
Autocatalytic Recombiners, water injection system (i.e., so-called Safety Injection System), and spray system (i.e., Containment Heat Removal System). The benchmark exercise introduces a stochastic character in the three main processes changing the dynamics: water injection, spray system, and \( \text{H}_2 \) combustion, but not in the recombiners as they act as passive systems, being always available during the transient. The general schema of considered processes during the considered accident is presented in Figure 5.

Both the stochastic occurrence of the processes (where time scale with time variables is presented in Figure 6) and its physical effect for the considered test case are specified (arbitrary assumed) as follows.

**Safety (Water) Injection System (SIS)**

*Time Activation/Actuation.* The SIS starts before the vessel rupture \( (t = 14220 \text{ s}) \), which is related to the end of core degradation consideration, when hydrogen mass released in containment can be maximal. No failure during operation of the system is considered. However, a probability of having the system available is indeed considered. If it occurs before the total core uncovering \( (t = 5875 \text{ s}) \) the situation is supposed to be safe (few hydrogen is produced and the vessel rupture is avoided). The probability of this situation is assumed to be 0.5. Between total core uncovering \( (t = 5875 \text{ s}) \) and vessel rupture \( (t = 14220 \text{ s}) \) the probability that water injection is available is assumed to be 0.5, as conditional probability, thus a total probability is equal to 0.25. In addition, the time when it does start follows a uniform PDF. Finally, it is assumed that all the injected water will entirely be changed by hydrogen.

**Effects.** The only direct assumed effect is an increase in the hydrogen flow rate coming from the primary system, thus, an increase in the number of moles of hydrogen.

**Containment Heat Removal (Spray) System (CHRS)**

*Time Activation/Actuation.* The probability that the CHRS can be activated after the core uncovers and before the vessel rupture \( (t = 14220 \text{ s}) \) is assumed to be equal to 0.5. If it can be activated, the time to start is uniformly distributed.

**Effects.** The actuation of this system produces condensation in the containment. Therefore, the direct effects of this system are the decrease of the number of moles of steam and a decrease of the temperature and the pressure. The spray system cannot be stopped. Once it starts, it continues to work,
so those effects have to be considered in the entire transient after the actuation.

Hydrogen Combustion

Time Activation/Actuation. Hydrogen combustion can occur only if the hydrogen concentration inside the containment is sufficient. There are defined two regions for initiating the combustion, the first due to the flammability conditions of the gases mixture and the second one due to the ignition capability of the recombiners. A variable delay is defined for each region. The delay before combustion depends on the hydrogen concentration inside the containment and is supposed to be shorter as molar fraction of hydrogen gets higher.

Effects. Hydrogen combustion is considered as a shock event, only causing sudden variations of some of the state variables. Therefore, combustion is assumed to happen instantaneously with a fraction of burnt hydrogen C given by an assumed uniform distribution between 0.05% and 100%.

Passive Autocatalytic Recombiners (PAR)

Time Activation/Actuation. They are supposed to be working at the beginning of the transient and never stop throughout the transient, so no random time is associated with this system of recombiners.

Effects. They have the similar effect as the combustion, that is, a decrease of the number of moles of hydrogen and oxygen.

Containment Failure Criteria. The containment failure occurs when there is overpressurization due to hydrogen combustion. In order to simplify the analysis of results, there were specified two different criteria for failure due to overpressurization. The initial criterion supposes that the containment fails with probability equal 1 if the pressure inside the containment after combustion exceeds a given value (containment pressure limit). The reference case for the pressure limit was fixed to 0.5 MPa. However finally, in Step-1 a more realistic assumption for containment failure has been considered, where the containment rupture probability is a routine based function of the amplitude of the pressure peak.

Scenario Definition. Five scenarios are determined by variation of different events related to the activation of considered safety systems. The occurrence of scenarios is based on the activation of the Safety Injection System and the Containment Heat Removal System (see Table 1).

The probability of each scenario occurrence is related to the activation of the safety systems described above. However, it is not possible to calculate analytically the precise theoretical estimates of scenario occurrence probability. The probabilities of the occurrences of the safety systems have discrete distribution; besides, scenarios C and E can be finished after the first event due to containment rupture.

The specification of the test case is supposed to be quite easy to implement in order to limit as far as possible effort of comparison with other calculations. For that reason, the proposed specifications and assumptions are rather analytical, to avoid direct use of complex severe accident codes and at least partly to allow a simplified analytical assessment taking into account timing of various events.

4.2 Accident Dynamics and Timing. List of used "physical" information specified according to the benchmark [26] is described as follows:

(i) A representative ASTEC transient without spray and reflooding
   (a) start and finish: beginning of core degradation, vessel rupture;
   (b) dynamic effect: hydrogen mass released in containment;
   (c) process: containment pressure as a function of time.

(ii) A simple law that allows predicting pressure evolution as a function of time after spray system start.

(iii) A simple law that allows predicting H₂ release after core reflooding.

(iv) A simple law that allows to predict recombiners efficiency in function of H₂ and H₂O concentrations.

(v) Criteria for hydrogen combustion, based on Shapiro diagram and effect of ignition by recombiners.

(vi) The probability of containment failure as a function of pressure peak.

(vii) Pressure peak in containment due to combustion evaluated by so called PAICC model (provided by IRSN).

In addition, the following assumptions were used: the atmosphere ignition within a short delay is very probable if the recombiners ignition criteria are achieved for average H₂ concentration and local (partial) and multiple ignition have been taken into account.

The scenario which reflects the accident dynamics and more realistic conditions for safety system actuation are actually not as simple as ordinary sequence of events. The specification of the hypothetical transient of the initial accident is related to the following events:

(1) The initiating event of reactor core dewatering (when core uncover starts).
(2) The core dewatering and vessel rupture.
Table 2: Time related variables and parameters.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>Time scale variable (equal to 0, when LoCA)</td>
<td>s</td>
</tr>
<tr>
<td>$t_d$</td>
<td>Time moment, when core dewatering starts; time moment, when core uncover starts; that is, core degradation process starts ($t_d = 4080$)</td>
<td>s</td>
</tr>
<tr>
<td>$h_d$</td>
<td>Time delay between LoCA and core dewatering ($h_d = t_d = 4080$)</td>
<td>s</td>
</tr>
<tr>
<td>$t_c$</td>
<td>Time moment, when clad oxidation starts ($t_c = t_d + h_c = 4125$)</td>
<td>s</td>
</tr>
<tr>
<td>$h_c$</td>
<td>Time delay between core dewatering and clad oxidation, that is, start of hydrogen production ($h_c = 45$)</td>
<td>s</td>
</tr>
<tr>
<td>$t_i$</td>
<td>Time moment, when total core uncover occurs ($t_i = 5875$ if $t_i &gt; t_d$)</td>
<td>s</td>
</tr>
<tr>
<td>$t_s$</td>
<td>Time moment, when water spray starts ($t_s = \infty$ if $t_s &gt; t_d$; $t_s = U(t_i, t_f)$)</td>
<td>s</td>
</tr>
<tr>
<td>$t_f$</td>
<td>Time moment, when containment fails ($t_f = \infty$ if $t_f &gt; t_d$; $t_f = \infty$ if $t_f &lt; t_c$)</td>
<td>s</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Time delay (core dewatering phase) between core dewatering and start of vessel rupture, that is, vessel damage</td>
<td>s</td>
</tr>
</tbody>
</table>

Note: $U(a, b)$: uniform distribution in interval $(a, b)$.

![Figure 7: Event tree for the test case scenarios.](image)

This transient corresponds to the following situation:

1. Loss of Coolant Accident (LOCA) with a $3''$ break size occurs on the cold leg of RCS.
2. The Safety Injection System (SIS or water injection system) and Containment Heat Removal System (CHRS or spray system) are not available until the beginning of core dewatering.
3. The steam generators are available but not used by the operators.
4. No water injection occurs before core dewatering.
5. The reactor is operating at nominal power before the initiating event.
6. The calculated core dewatering occurs at 4080 s (1h 08 min). The vessel rupture occurs at 14220 s (3h 57 min) if no action is undertaken.

If no action is taken, the time scale of this transient is illustrated in Figure 6 and time related variables and parameters are described in Table 2.

During the core dewatering phase, the situation is supposed to be as follows:

1. A water injection means is available (with an average flow rate = 0.833 kg/s) and can be used by the operators.
2. The spray system (CHRS) is available and can be used by the operators.
3. Water injection after the beginning of clad oxidation causes an increase of the hydrogen flow rate towards containment.
4. Hydrogen combustions can occur if the containment gas mixture is flammable; recombiners, because of their high temperature, can initiate a combustion; such combustions can be total (all the hydrogen in the containment is burnt) or not.

4.3. Event Tree and Simplified Modelling. In relation to dynamic effects in the test case, there is possibility to construct a simple event tree (see Figure 7) and to relate all sequences to the previously described scenarios (see Table 1). There is possibility to note that in this event tree the time-dependent sequence of two safety systems are reflected as the order of these systems is also considered in scenario definition.

As the test case is relatively simple and the dynamic effects are limited, the initial classical PSA based solution will allow
a comparison with the results obtained by SDTPD (see the next sections) and a discussion about the advantages and disadvantages of both approaches.

Having event tree constructed, it is quantified. For each branch, its probability is calculated and, then, the conditional rupture probability (given the considered scenario) is estimated. At the end, the total probability of the containment rupture is calculated by combining these results. There is also need to note that the event tree and values calculated and shown below are related to the Step-1 of the test case and to the rupture limit defined as pressure equal to 5 bar (without any stochastic variation).

According to the initial assumptions, the probability $P_{\text{SIS}^-}$ that water is injected before the total core uncovers ($t_i = 5875$ s) is equal to 0.5. And the probability $P_{\text{SIS},+}$ that this system does not start between the total core uncovering and the end of the transient is equal to 0.25. Thus, the probability that water is injected $P_{\text{SIS}} = 0.75$. In addition, it was assumed that the probability to start CHRS is $P_{\text{CHRS}} = 0.5$.

During A scenario nothing happens, that is, no water injection and no spray system actuation. Thus, the probability of this type of scenario can be estimated as follows:

$$P_A = P_{\text{SIS}} \cdot P_{\text{CHRS}} = (1 - P_{\text{SIS}}) \cdot (1 - P_{\text{CHRS}}) = 0.125,$$

where $P_{\text{SIS}}$ is the probability that no water is injected; that is, SIS does not start, and $P_{\text{CHRS}}$ is the probability that CHRS does not start.

During B scenario there is no water injection, but CHRS starts. Thus, the probability of this type of scenario can be expressed and estimated as follows:

$$P_B = P_{\text{SIS}} \cdot P_{\text{CHRS}} = (1 - P_{\text{SIS}}) \cdot P_{\text{CHRS}} = 0.125.$$

During C scenario, CHRS (at time moment $t_i$) starts before water is injected (at time moment $t_j$). According to the initial assumptions and taking into account time related variables and parameters described in Table 2, this scenario reflects one of the following three situations with the corresponding probabilities estimated:

1. CHRS and SIS start before the total core uncovering occurs ($t_i = 5875$), with probability $P_{C,1}$:

$$P_{C,1} = \int_{t_d}^{t_i} \int_{t_j}^{t_i} P_{\text{SIS},-} \cdot dt_i \cdot dt_j = 0.0221.$$

2. CHRS starts before $t_i = 5875$ and SIS starts after this time moment $t_j$, with probability $P_{C,2}$:

$$P_{C,2} = \int_{t_d}^{t_i} \int_{t_j}^{t_i} P_{\text{CHRS}} \cdot dt_i \cdot dt_j = 0.0221.$$

3. CHRS and SIS start after the total core uncovering occurs ($t_j = 5875$), with probability $P_{C,3}$:

$$P_{C,3} = \int_{t_i}^{t_j} \int_{t_j}^{t_i} P_{\text{SIS},+} \cdot dt_i \cdot dt_j = 0.0516.$$

Thus, the probability of C scenario, including all three situations, finally can be estimated as follows:

$$P_C = P_{C,1} + P_{C,2} + P_{C,3} = 0.096.$$  \hspace{1cm} (11)

During D scenario initially SIS starts and CHRS does not start. According to the initial assumptions and taking into account time related variables and parameters described in Table 2, this scenario reflects one of the following two situations with the corresponding probabilities estimated:

1. SIS starts before $t_i = 5875$, with probability $P_{D,1} = P_{\text{SIS},-} = 0.5$.

2. SIS starts after $t_i = 5875$ and CHRS does not start, with probability $P_{D,2}$:

$$P_{D,2} = P_{\text{SIS},+} \cdot \int_{t_i}^{t_r} \frac{(1 - P_{\text{CHRS}})}{t_r - t_d} \cdot dt_i \approx 0.1029.$$  \hspace{1cm} (12)

Thus, the probability of D scenario, including both situations, finally can be estimated as follows:

$$P_D = P_{D,1} + P_{D,2} = 0.603.$$  \hspace{1cm} (13)

During E scenario, SIS (at time moment $t_i$) starts before the start of CHRS (at time moment $t_j$). So, water has to be injected after $t_i = 5875$. Indeed, if water is injected before $t_i = 5875$, the scenario is stopped and the spray system cannot start (this situation corresponds to type D). Thus, the probability of this type of scenario can be expressed and estimated as follows:

$$P_E = \int_{t_i}^{t_r} \frac{P_{\text{SIS}}}{t_r - t_d} \cdot dt_i \int_{t_i}^{t_r} P_{\text{CHRS}} \cdot dt_j = 0.052.$$  \hspace{1cm} (14)

On the basis of the analytically estimated scenario probabilities and taking into account possible transitions from branch E to branch D, they can be finally presented in Table 3.

Due to assumptions for simplification in analytical modelling the estimate of each scenario probability does not fully correspond to the possible reality because the calculations do not take into account the occurrence of combustions leading to rupture between the first event and the second one. For example, if SIS is the first system to start, there are three different situations after this start:

1. Combustion occurs and leads to a rupture (in spite of other conditions).

2. Combustion occurs but does not lead to a rupture and then CHRS starts.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Scenario probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.125</td>
</tr>
<tr>
<td>B</td>
<td>0.125</td>
</tr>
<tr>
<td>C</td>
<td>0.096</td>
</tr>
<tr>
<td>D</td>
<td>0.603</td>
</tr>
<tr>
<td>E</td>
<td>0.052</td>
</tr>
</tbody>
</table>

Table 3: Analytically estimated scenario probabilities.
### Table 4: Uncertain conditions and parameters.

<table>
<thead>
<tr>
<th>Number</th>
<th>Full parameter name</th>
<th>Reference</th>
<th>Distribution</th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of moles of steam</td>
<td>1.72E + 06</td>
<td>Uniform</td>
<td>1383240</td>
<td>2034004</td>
</tr>
<tr>
<td>2</td>
<td>Temperature inside the containment</td>
<td>382.05</td>
<td>Uniform</td>
<td>369.75</td>
<td>383.55</td>
</tr>
<tr>
<td>3</td>
<td>Pressure inside the containment</td>
<td>2.28</td>
<td>Uniform</td>
<td>2</td>
<td>2.5</td>
</tr>
<tr>
<td>4</td>
<td>Factor of spray efficiency to condensate hydrogen</td>
<td>1</td>
<td>Uniform</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>Factor of hydrogen mass flow rate in the range 1: [5875; 7000] or [9401; 14220]</td>
<td>0.5</td>
<td>Uniform</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Factor of hydrogen mass flow rate in the range 2: [7001; 9400]</td>
<td>0.5</td>
<td>Uniform</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Factor of water injection (SIS) availability</td>
<td>0.5</td>
<td>Uniform</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Factor of spray injection (CHRS) availability</td>
<td>0.5</td>
<td>Uniform</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Coefficient $K_1$ of recombiner hydrogen flow rate</td>
<td>0.003</td>
<td>Uniform</td>
<td>0.0015</td>
<td>0.0045</td>
</tr>
<tr>
<td>10</td>
<td>Coefficient $K_2$ of recombiner hydrogen flow rate</td>
<td>0.0037</td>
<td>Uniform</td>
<td>0.00185</td>
<td>0.00555</td>
</tr>
<tr>
<td>11</td>
<td>Factor of peak pressure to cause a rupture</td>
<td>0.5</td>
<td>Bernoulli (0.5)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Activation of spray at 2.4 MPa pressure limit</td>
<td>0</td>
<td>Bernoulli (0.5)</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

(3) Combustion does not occur as the CHRS starts before any combustion.

The first situation may correspond to scenario D while the second and third situations may correspond to scenario E.

In general, the best estimates probabilities of various scenarios, which in some cases lead to containment rupture, in simulation depend on timing and stochastic parameters. The probabilities of combustion and containment rupture cannot be expressed analytically but may be estimated performing simulations during which any probability of scenario is also estimated. The difference between the analytically estimated probability and the simulation based estimates depends on the precision of the time-dependent reality and uncertainty reflection in the simulations.

### 4.4. Simulation of Time-Dependent and Stochastic Rupture.

The model for calculations was prepared using SDTPD approach and various simulation and analysis software. The physical model was based on various deterministic laws and process variables (like hydrogen flow rate, number of moles of hydrogen, steam and oxygen, temperature and pressure inside the containment, etc.) and combustion phenomena modelling, all together combined with simulation of various stimuli and corresponding delays, dynamics and random events related to the Passive Autocatalytic Recombiners (PAR), Safety Injection System (SIS), and Containment Heat Removal System (CHRS) [26]. The considered containment physical phenomena were mostly related to the containment gas phase and dynamic hydrogen combustions. The pressure and temperature peaks for certain conditions were calculated by the so-called PAICC model (provided by IRSN). The developed time-dependant hybrid (continuous-discrete) physical model was used to perform 100,000 runs for each simulation case.

The final task of simulations was to estimate the containment rupture probability and uncertainty. Different scenarios were separated and analyzed depending on the possible failures of safety systems and related consequences. Using the developed probabilistic model, many histories were simulated with respect to the timing and grouped to the sequences of dynamics as grouped paths of processes [16]. Later on, the sequences of dynamics were grouped depending on the scenarios considered. For each scenario, the estimate of the conditional rupture probability was calculated.

In general, the result of probabilistic simulation is affected by internal stochastic variables of the probabilistic model with deterministically defined dynamics. However, during the simulation, the result also depends on values of uncertain input (see Table 4). For instance, the following assumption and coefficients $K_1$ and $K_2$ were used for flow rate of recombiner hydrogen (and oxygen) calculation. The mass flow rate (g/s) of hydrogen recombined by the PAR system was specified by the following law: $Q_{H_2}^{rec} = (K_1 \cdot P + K_2) \cdot [H_2]$, with $[H_2] = N_{H_2} / N_{tot} \cdot 100$ and $N_{tot} = N_{steam} + N_{O_2} + N_{H_2} + N_{H_2}$ (number of moles). For Step-1: $K_1 = 3.0 \text{ g/s/bar}$, $K_2 = 3.7 \text{ g/s}$ and for Step-2: $K_1$ and $K_2$ have an uncertainty band between 0.5 and 1.5 (uniform distribution) times the above reference values. The mass flow rate (kg/s) for recombiner oxygen is eight times the mass flow rate for hydrogen.

The difference in the results and impact of uncertain parameters on the results and the importance of events, which
A result of one probabilistic simulation with many runs (simulation of histories) is an estimate of the considered final event frequency (i.e., containment rupture probability estimate) for the specific set of fixed input parameters. Using a result based on point input values, it is not possible to estimate the distribution as there is only one result received from one probabilistic simulation with the specified amount of runs (e.g., 10,000 histories); however, it is possible to express the deviation of this result due to the probabilistic model itself.

Having more runs, the result of the probabilistic simulation will be more precise. However, with the same input and the same amount of runs, it is expected that the result of each simulation will be slightly different. This variation does not depend on the possible uncertainty of inputs. In this case, the distribution and statistical characteristics (e.g., mean and standard deviation) of the result depends only on the probabilistic model and can be based on statistical analysis of result deviations from simulations with the same inputs and the same amount of runs.

One run or history from simulation, taking into account a single set of model inputs, only treats one possible final event (e.g., containment rupture), which influences the result of the whole probabilistic simulation with various histories. Using one run, it is not possible to estimate failure frequency (i.e., the considered result) for one specific set of inputs.

Using a lot of (e.g., 10,000) probabilistic simulations based on one run with different set of inputs, an averaged influence of uncertain inputs can be considered. However, in such case, it is still not possible to analyze the sensitivity of the uncertain result with respect to the uncertainty of inputs.

In case the uncertainty and sensitivity of the result is considered, one must get a distribution of results based on a lot of runs (e.g., 10,000 histories) for each simulation (B option in Figure 8). For instance, 100 simulations with different sets of uncertain inputs will give the distribution of 100 results. In this case, the uncertainty of the result is related to the uncertain inputs and it is possible to perform the sensitivity of uncertainty analysis (SUA).

The last option, whose results are presented below, means that random start of CHRS at 2.4 MPa pressure was considered (i.e., automatic start of spray injection is working with probability equal to 0.5). The analysis itself is related to the 100 probabilistic simulations with the set of different uncertain parameters for each simulation but the same parameters for all histories in the each simulation case.

4.5. Analysis of Uncertain Rupture Frequency. As in the test case, the sample size of model results was 100 (i.e., results after 100 simulations with 10,000 histories each), it was possible to evaluate the (0.95, 0.95) tolerance interval. While the upper limit of rupture frequency is 0.157 and the lower limit is 0.000, with 0.95 probability it is possible to predict that 95 percent of model outputs (rupture frequencies) are in interval (0.000, 0.157). In addition, estimating the median, which is equal to 0.013, it is possible to say that half of the model results did not exceed the value 0.013. The variation of 100 results can be seen from Figure 9 with frequencies of combustions and frequencies of possibly following ruptures.

There is observable correlation in the variation of combustion frequency and rupture frequency and it is possible to note that some results of simulations are with very high combustion frequency. In some cases, where the combustion frequency is around 1, in each history, there is possibility to expect one or more combustions. In all simulations there are some histories with ruptures. However, in some histories there are no ruptures or even combustions at all.

In addition, it is possible to note that some sets of model parameters give quite large deviations in the rupture frequencies. The sample mean of rupture frequency is 0.0228, the standard deviation is 0.0277, the 5th quantile is 0.0004, and the 95th quantile is 0.0727.

The statistical characteristics (mean and standard deviation) obtained using the results from different probabilistic simulations (taking correspondingly different computing time) are presented in Table 5. It is possible to note that
Table 5: Simulation results for Step-2.

<table>
<thead>
<tr>
<th>Number of histories</th>
<th>Rupture frequency</th>
<th>Standard deviation</th>
<th>Combustion frequency</th>
<th>Standard deviation</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 \times 10,000$</td>
<td>0.022828</td>
<td>0.027757</td>
<td>0.17787</td>
<td>0.20790</td>
<td>9 min 00 s</td>
</tr>
<tr>
<td>$100 \times 1,000$</td>
<td>0.02284</td>
<td>0.027702</td>
<td>0.17688</td>
<td>0.20852</td>
<td>1 min 04 s</td>
</tr>
<tr>
<td>$100 \times 100$</td>
<td>0.0219</td>
<td>0.030804</td>
<td>0.16860</td>
<td>0.19789</td>
<td>0 min 11 s</td>
</tr>
</tbody>
</table>

This also points out the fact that significant hydrogen concentration could be reached, leading to a flammable gas mixture, but, because of the presence of high steam concentrations, hydrogen burn may be prevented. If the atmosphere is undergoing rapid condensation, for example, by spray initiation, a potentially detonable mixture could form rapidly in case of a high concentration of hydrogen. Additionally, depressurization can also take place as a result of some exchange taking place within the atmosphere, as in the case of energy and mass exchange with containment sprays (e.g., in B, C, and E scenarios).

The probabilities of various scenarios (for nominal and uncertain parameters) and related sensitivity measures (analysing these probabilities of scenarios) are presented in Figure 12.

In the future, these scenarios (see Table 1) probabilities and sensitivity measures could be also used for the additional simulations in order to focus on the rarest sequences with the quite severe consequences and reduce the computation time of the simulations performed with SDTPD.

Having the probabilities of scenarios, the weight of each scenario is known. It depends on the frequency of events which corresponds to the scenario and can be changed by increasing the values of the parameters with the highest positive measures of sensitivity. By increasing uncertainty for these parameters for the rarest scenario there is possibility to increase the chance during the simulation to focus on the sequences with less probable conditions but possibly more severe consequences. The different parameters impact on the severity may also be reflected by other sensitivity measures, which express parameters effects on the rupture probability.

Similarly it is possible to have more scenarios or consider sequence grouping with respect to sensitivity measures and vice versa. Hence, it is possible to identify and force rare accidental sequences. After complete simulations of these rare sequences, the results obtained can be weighted and incorporated in the other results in order to have the global probability of the uncertain events. In addition, this idea can be extended applying a time-dependent sensitivity analysis, which may guide the generation of events in the sequence. Then, the computation time may also be reduced, while keeping possibility to identify some scenarios likely to be forgotten by basic Monte Carlo simulation due to their very low probability. A time-dependent analysis was performed for the rupture frequency. While scalar analysis considers only final simulation results, the time-dependent analysis can reveal uncertainty and sensitivity of the model at any moment of the modelling process. The time-dependent analysis complements the results given by the scalar analysis. Most of the
rupture probability estimates in time lie under 0.1, but some particular sets of model parameters give higher values (e.g., one, which is related to maximum rupture frequency).

Time-dependent uncertainty and sensitivity analysis can show how model parameters affect the result at each simulation moment (see Figures 13 and 14).

The set of considered parameters can also be extended and related to the timing; that is, the set of input parameters can reflect uncertain delays and stochastically distributed time moments of events occurrence. However, this was not a case in the Step-2 of the benchmark exercise. For such an investigation, the extended version of the benchmark exercise could be performed in the future. The additional uncertainty and sensitivity analysis could be also useful in order to show how the uncertain timing is important for the modelling and analysis of a dynamic system trying to make it more robust and reliable.

5. Discussions and Conclusions

Even complex systems behaviour and accidental transients involved in level-1 PSA are usually treated as mainly having been governed by the failure or the success of systems functions or by well prescribed human actions. The chance of occurrence of various events is relatively well known and they are ordered in a chronological way. Consequently, the classical methods based on these assumptions and used in level-1 PSA, the event and fault trees, have been widely developed.

But the situation is very different in case of the time-dependent hybrid systems and accidental transients involved in level-2 PSA. Indeed, these systems and accidents are mainly governed by physical phenomena and dynamics aspects. It is impossible to predict the chronology of the events and each event cannot be reduced to a binary situation
(failure or success) but combined and functional aspects have to be considered. Classical methods developed for level-1 PSA do not cope with all the specificities of probabilistic dynamics in level-2 PSA and there is currently no single method devoted and specially adapted to the related issues of severe accidents. Nevertheless, the first attempts (related to memoryless Markov processes) have been made on the basis of the event trees, trying to include some dynamics and uncertainty aspects. They constitute a first step but are not really a satisfactory solution (when each state depends only on previous state). The most promising attempts are based on dynamic reliability methods because they explicitly model the dynamic evolution of the system, the time-dependent aspects of this evolution and take into account the interactions between the system states, the physical variables values, and the human actions.

Recently, the TSD as an extension of the dynamic reliability and simplification of SDTPD has been developed. This is a practical approach of general theory based on dynamic reliability, supplemented by the notion of stimulus and delay. Hence, each physical event is divided into two phases: the stimulus activation (as soon as all the conditions necessary to the event occurrence are met) and delay (before the actual occurrence of the event), with a possible stimulus deactivation if the conditions are no longer met (cancelled). It allows modelling in an accurate way the interaction between events, which is one of the advantages of the TSD (in comparison to other methods based on the Markov processes).

For example, let us consider the hydrogen issue one more time. Hydrogen is released in the containment atmosphere and the gases mixture becomes flammable. There is no heat source and, consequently, no combustion. So, nothing happens. But the system continues evolving and the recombiners start (eventually with delay). Consequently, they reduce the amount of hydrogen in the atmosphere and the hydrogen concentration decreases; the gases mixture is no longer flammable and no combustion is possible, even in case of a spark. This kind of situations with events interactions is impossible to model or adapt with other classical techniques or methodologies.

Considering the drawbacks of time-dependent reliability and uncertainty analysis, the concept of dynamic reliability and stimulated dynamics was applied for the analysis of hybrid systems with uncertain events.

As shown in the test case, the time step (taking correspondingly different computing time) influence on the results is very limited. Consequently, it is possible to reduce the computing time by using a larger time step or smaller numbers of histories, without meaningful change of the containment rupture frequency mean and standard deviation, which is quite high due to uncertain parameters.

The stimulated dynamics with the considered uncertainty and sensitivity analysis allows a detailed simulation and representation of the dynamic system uncertainty. Taking into account which parameters mostly influence the uncertainty of the system, such simulation can be used to search for rare, but possibly severe, conditions of a dynamic system.

The combinations of uncertain events and process values can lead to various conditions or situations and probabilistic results can change. The impact of these uncertainties does not only concern the final results but also could lead to the totally new situations. Consequently, further investigations can be carried out.

The considered techniques of modelling clearly allow for a systematic analysis of complex system reliability and uncertainty. The developed approach for analysis of hybrid systems with uncertain events can be efficiently used to estimate system failure probability or reliability and at the same time to analyze the uncertainty of this estimate.

**Conflict of Interests**

The author declares that there is no conflict of interests regarding the publication of this paper.

**Acknowledgments**

The author would like to acknowledge the contacts started with the PSA2 group in the Severe Accident Research Network of Excellence (SARNET). The author also thanks Pierre-Etienne Labeau, Agnes Peeters, and Tadas Eimontas, for their valuable input via numerous discussions and participation in the related analysis. This research was partly funded by a grant.
from the Fonds de la Recherche Scientifique in Belgium and Research Council of Lithuania.

References

Research Article

Demonstration of Emulator-Based Bayesian Calibration of Safety Analysis Codes: Theory and Formulation

Joseph P. Yurko,¹,² Jacopo Buongiorno,¹ and Robert Youngblood³

¹MIT, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
²FPoliSolutions, LLC, 4618 Old William Penn Highway, Murrysville, PA 15668, USA
³INL, P.O. Box 1625, Idaho Falls, ID 83415-3870, USA

Correspondence should be addressed to Joseph P. Yurko; jyurko@fpolisolutions.com

Received 16 January 2015; Revised 1 April 2015; Accepted 28 May 2015

Academic Editor: Francesco Di Maio

Copyright © 2015 Joseph P. Yurko et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

System codes for simulation of safety performance of nuclear plants may contain parameters whose values are not known very accurately. New information from tests or operating experience is incorporated into safety codes by a process known as calibration, which reduces uncertainty in the output of the code and thereby improves its support for decision-making. The work reported here implements several improvements on classic calibration techniques afforded by modern analysis techniques. The key innovation has come from development of code surrogate model (or code emulator) construction and prediction algorithms. Use of a fast emulator makes the calibration processes used here with Markov Chain Monte Carlo (MCMC) sampling feasible. This work uses Gaussian Process (GP) based emulators, which have been used previously to emulate computer codes in the nuclear field. The present work describes the formulation of an emulator that incorporates GPs into a factor analysis-type or pattern recognition-type model. This “function factorization” Gaussian Process (FFGP) model allows overcoming limitations present in standard GP emulators, thereby improving both accuracy and speed of the emulator-based calibration process. Calibration of a friction-factor example using a Method of Manufactured Solution is performed to illustrate key properties of the FFGP based process.

1. Introduction

Propagating input parameter uncertainty for a nuclear reactor system code is a challenging problem due to often non-linear system response to the numerous parameters involved and lengthy computational times, issues that compound when a statistical sampling procedure is adopted, since the code must be run many times. Additionally, the parameters are sampled from distributions that are themselves uncertain. Current industry approaches rely heavily on expert opinion for setting the assumed parameter distributions. Observational data is typically used to judge if the code predictions follow the expected trends within reasonable accuracy. All together, these shortcomings lead to current uncertainty quantification (UQ) efforts relying on overly conservative assumptions, which ultimately hurt the economic performance of nuclear energy.

This work adopts a Bayesian framework that allows reducing computer code predictive uncertainty by calibrating parameters directly to observational data; this process is also known as solving the inverse problem. Unlike the current heuristic calibration approach, Bayesian calibration is systematic and statistically rigorous, as it calibrates the parameter distributions to the data, not simply tune point values. With enough data, any biases from expert opinion on the starting parameter distributions can be greatly reduced. Multiple levels of data are easier to handle as well, since Integral and Separate Effect Test (IET and SET) data can be used simultaneously in the calibration process. However, implementing Bayesian calibration for safety analysis codes is very challenging. Because the posterior distribution cannot be obtained analytically, approximate Bayesian inference with sampling is required. Markov Chain Monte Carlo (MCMC) sampling algorithms are very powerful and have become increasingly widespread over the last decade [1]. However, for even relatively fast computer models practical implementation of Bayesian inference with MCMC would simply take too long because MCMC samples must be drawn in series.
As an example, a computer model that takes 1 minute to run but needs $10^5$ MCMC samples would take about 70 days to complete. A very fast approximation to the system code is thus required to use the Bayesian approach. Surrogate models (or emulators) that emulate the behavior of the input/output relationship of the computer model but are computationally inexpensive allow MCMC sampling to be possible. An emulator that is 1000x faster than the computer model would need less than two hours to perform the same number of MCMC samples. As the computer model run time increases, the surrogate model becomes even more attractive because MCMC sampling would become impractically lengthy.

Gaussian Process- (GP-) based emulators have been used to calibrate computer code for a variety of applications. Please consult [2–5] for specific cases as well as reviews of other sources. This work applies a relatively new class of statistical model, the function factorization with Gaussian Process (FFGP) priors model, to emulate the behavior of the safety analysis code. The FFGP model builds on the more commonly used GP emulator but overcomes certain limiting assumptions inherent in the GP emulator, as will be explained later. The FFGP model is therefore better suited to emulate the complex time series output produced by the system code. The surrogate is used in place of the system code to perform the parameter calibration, thereby allowing the observational data to directly improve the current state of knowledge.

The rest of this paper is organized as follows. An overview of the entire emulator-based Bayesian calibration process is described in Section 2. Section 3 discusses the emulators in detail. The first half of Section 3 summarizes the important expressions related to GP emulators. Most of these expressions can be found in numerous other texts and references on GP models, including [6, 7]. They are repeated in this paper for completeness as well as providing comparison to the FFGP expressions in the latter half of Section 3. Section 4 presents a method of manufactured solutions-type demonstration problem that highlights the benefits of the FFGP model over the standard GP model.

2. Overview of Emulator-Based Bayesian Calibration

As already stated, the emulator-based approach replaces the potentially very computationally expensive safety analysis code (also known as a simulator, computer code, system code, or simply the code) with a computationally inexpensive surrogate. Surrogate models are used extensively in a wide range of engineering disciplines, most commonly in the form of response surfaces and look-up tables. Reference [4] provides a thorough review of many different types of surrogate models. The present work refers to the surrogates as emulators to denote that they provide an estimate of their own uncertainty when making a prediction [5]. An emulator is therefore a probabilistic response surface which is a very convenient approach because the emulator’s contribution to the total uncertainty can be included in the Bayesian calibration process. An uncertain (noisy) emulator would therefore limit the parameter posterior precision, relative to calibrating the parameters using the long-running computer code itself. Obviously, it is desirable to create an emulator that is as accurate as possible relative to the computer code, which limits the influence of error and uncertainty on the results.

The emulator-based approach begins with choosing the input parameters and their corresponding prior distributions. If the emulator was not used in place of the system code, the Bayesian calibration process would start in the exactly same manner. The priors encode the current state of knowledge (or lack thereof) about each of the uncertain input parameters. Choice of prior for epistemically uncertain variables is controversial and relies heavily on expert opinion. Justification for the priors used in the applications of this work is given later on, but choice of the priors is not the focus of this work. Additionally, the choice of the specific input parameters to be used for calibration may be controversial. Dimensionality reduction techniques might be used to help screen out unimportant input parameters [4]. Some screening algorithms such as the Reference Distribution Variable Selection (RDVS) algorithm use GPs to identify statistically significant input parameters [8]. In the nuclear industry specifically, expert opinion-based Phenomena Identification and Ranking Tables (PIRTs) are commonly used to down-select the most important physical processes that influence a Figure of Merit (FOM) [9]. More recently, Quantitative PIRTs, or QPIRTs, have been used in place of the traditional expert opinion PIRTs to try to remove bias and to capture relevant physical processes as viewed by the computer code [10, 11]. No matter the approach, the set of input parameters and their corresponding prior distribution must be specified.

In the emulator-based approach, the prior has the additional role of aiding in choosing the training set on which the emulator is based. As the phrase implies, the training set is the sequence of computer code evaluations used to build or train the emulator. Once trained on selected inputs and outputs, the emulator reflects the complex input/output relationship, so training is clearly an essential piece of the emulator-based approach. There are numerous methods and decision criteria for the selection of the training set; see [4, 5] for more details. Reference [12] provides an excellent counterpoint for the dangers of not using enough points in generating the training set. This work does not focus on choosing the “optimal” or “best” training set, which is an active area of research. The input parameter prior is used to set bounds on the input parameter values; Latin Hypercube Sampling (LHS) is then used to create a “space filling” design within those bounds. Although not guaranteed to produce the best possible training set, this method adequately covers the prior range of possible input parameter values. An active area of research is how to enhance the training set during the calibration process itself, in order to focus more on the posterior range of possible values.

With the training input values chosen, the computer code is run the desired number of times to generate the training output. The complete training set is then the training input values with their corresponding training output. The emulator is then built by learning specific characteristics that allow the emulator to represent the input/output relationship encoded in the training set. The specific characteristics that must be learned depend on the type of emulator being used.
Training algorithms for the standard GP emulator and FFGP emulator are described in Section 3.

Once trained, the emulator is used in place of the computer code in the MCMC sampling via an emulator-modified likelihood function. The modified likelihood functions are presented in Section 3 for each of the emulators used in this work. Regardless of the chosen type of emulator, the emulator-based calibration process results in uncertain input parameter posterior distributions and posterior-approximated predictions, conditioned on observational data. A flow chart describing the key steps in the emulator-based Bayesian calibration process is shown in Figure 1.

The emulator-based Bayesian calibration process presented in this work fixes the emulator once it is trained. Alternatively, the emulator could be constructed simultaneously with the calibration of the uncertain input parameters [2, 3]. The key difference between the two approaches is that the emulator-modified likelihood function in [2, 3] is not fixed since the emulator is not fixed. Formally, the alternative approach bases the emulator-modified likelihood function around the emulator prior predictive distribution whereas the work presented in this paper bases the emulator-modified likelihood function around the emulator posterior predictive distribution. The difference between the posterior and prior predictive distributions is described in detail in Section 3.2.4. The alternative approach therefore makes emulator predictions conditioned on both the training data and the observational data simultaneously. In some sense, the alternative approach is more of a data or information “fusion” method rather than a calibration-focused approach. The drawback of the alternative “data fusion” approach is that the emulator is not built until after the entire Bayesian calibration process is complete. Thus, if multiple levels of data such as from IETs and SETs are present, the emulators for all of the IETs and SETs must be calibrated simultaneously, which considerably complicates and slows the MCMC sampling. For those reasons, this work does not use the “data fusion” approach but fixes the emulator before starting the calibration of the uncertain input parameters.

3. Gaussian Process-Based Emulators

3.1. Overview. The emulators used in this work are based on Gaussian Process (GP) models and are considered Bayesian nonparametric statistical models. Nonparametric models offer considerably more flexibility than parametric models because the input/output functional relationship does not have to be assumed a priori by the user. The training data dictates the input/output relationship, just as a look-up table functions. As stated earlier, the emulator is a probabilistic model; therefore, the emulators are essentially probabilistic look-up tables. Nonparametric models are however considerably more computationally intensive than parametric models, because the training data is never discarded. If a large number of training runs are required to accurately capture the input/output trends, a nonparametric model might be considerably slower to run than a parametric model of the same data (e.g., a curve that fits the data).

The underlying principles of the GP model were developed in the 1960s in the geostatistics field where it was known as Kriging [4]. Since then Kriging has been widely used for optimization, but starting in the late 1980s and early 1990s, [13–15] popularized the approach as Bayesian approximations to deterministic computer codes. In the early 2000s, Kennedy and O’Hagan used the GP model to facilitate Bayesian calibration of computer codes [16]. Their work served as the foundation for this paper and many of the references cited in the previous section.

The machine learning community has also extensively used GP models for both regression and classification (regression is used for continuous functions while classification is used for discrete data) [6, 7]. Even with all of their flexibility, GP models are still somewhat limited by certain underlying assumptions to be discussed later, as well as the limitation in handling very large datasets (just as with any nonparametric model). In order to overcome these limitations and handle more complicated input/output relationships, many different approaches have been developed [6]. One such approach is based on combining GP models with factor analysis techniques; this is referred to as Gaussian Process Factor Analysis (GPFA) models [17, 18]. The work presented here uses the factor analysis based approach in order to handle very large datasets following the formulation of Schmidt [17].

3.2. Standard Gaussian Process (GP) Emulators

3.2.1. Formulation. Within the Bayesian framework, a Gaussian Process (GP) prior is placed on the computer code’s unknown output. The computer code, such as RELAP, is actually deterministic, meaning that the same output will result if the same input parameters and settings are used over
and over. The output, however, is in some sense unknown until the computer code is run, and it will therefore be treated as a random variable. A GP is a collection of random variables, any finite number of which have a jointly Gaussian distribution [6]. A Gaussian Process is simply a multivariate normal (MVN) distribution and is used presently as a prior distribution on the computer code input/output functional relationship.

The input \( x \) will be all \( D \) inputs to the computer code that the GP is trying to emulate: \( x = [x_1, x_2, \ldots, x_d, \ldots, x_D]^{\top} \). The superscript \( T \) denotes the transpose of the vector. The output, \( f(x) \), as stated above, is considered a random variable. A GP is completely specified by its mean function and covariance function. The mean function \( m(x) \) and covariance function \( k(x, x') \) are defined as [6]

\[
\begin{align*}
m(x) &= \mathbb{E} [f(x)], \\
k(x, x') &= \mathbb{E} [(f(x) - m(x))(f(x') - m(x'))]. \\
\end{align*}
\]

The GP is then defined as

\[
f(x) \sim \mathcal{GP}(m(x), k(x, x')).
\]

An important aspect of (2) is that the covariance between the outputs is written only as a function of the inputs. This is a key assumption in the simplicity of standard GP models, since all the covariance between two outputs depends only on the values of the inputs that produced those two outputs.

Following [6], as well as many other sources, the mean function is usually taken to be zero. Besides being the simplest approach to use, a zero mean function gives no prior bias to the trend in the data, since no mean trend is assumed. Covariance functions themselves depend on a set of hyperparameters; therefore, even though the GP is a nonparametric model, these hyperparameters specify the covariance function and must be learned from the training data. However, the GP model is still considered a nonparametric model, because a prediction still requires regressing the training dataset. Numerous covariance functions exist, ranging from very simple forms to very complex neural net-like functions [6]. Different forms have various advantages/disadvantages for different datasets, but the most common type used in the literature is the squared-exponential (SE) covariance function. The SE covariance function is usually parameterized as

\[
k(p, q) = \sigma_f^2 \exp \left( -\frac{1}{2} (p - q)^{\top} M (p - q) \right),
\]

where the subscripts \( p \) and \( q \) denote (potentially) two different values for the \( D \)-dimensional input vector \( x \). The hyperparameters in (3) are the signal variance \( \sigma_f^2 \) and the matrix \( M \), which is a symmetric matrix that is usually parameterized as a diagonal matrix:

\[
M = \text{diag}(l)^{-2}.
\]

Each diagonal element of \( M \) is a separate hyperparameter, \( l_d \), which serves as the characteristic length scale for the \( d \)-th input. Loosely speaking, the length scale represents how far the input value must move along a particular axis in input space for the function values to become uncorrelated [6]. Since each input parameter has its own unique length scale, this formulation implements what is known as automatic relevance determination (ARD), since the inverse of the length scale determines how relevant that input is. If the length has a very large value, the covariance will become almost independent of that input. Linkletter et al. [8] used ARD to screen out unimportant inputs using GP models.

Strictly speaking, the GP model can interpolate the training data exactly if no noise is allowed between the training data and the GP prior. However, implementation of an interpolating GP model might be difficult due to ill-conditioning issues [5, 6], which will be discussed later on. Allowing some hopefully very small noise between the GP prior and training data removes the numerical issues and turns the model into a GP regression (GPR) model. The GP prior is therefore actually placed on a latent (hidden) function, \( f(x) \), that must be inferred from the noisy data \( y \) [6]. This viewpoint brings to light the signal processing nature of the GPR framework, since the latent function is the true signal that must be inferred from the noisy data. In emulating computer codes, the training output is not noisy, but this setup provides a useful mathematical framework. The computer model output of interest, \( y \), is then related to the GP latent function \( f(x) \) as

\[
y = f(x) + \epsilon,
\]

where \( \epsilon \) is the error or noise. The error can take a variety of forms, but if a Gaussian likelihood model is used with independent and identically distributed (IID) noise, with zero mean and variance \( \sigma_n^2 \), the remaining calculations are all analytically tractable. More complicated likelihood models can be used and are often required to handle very complex datasets, but the remaining calculations would no longer have analytical expressions.

At this point, some important notation needs to be defined. If there are a total of \( N \) training points, the inputs are stacked into an \( N \times D \) matrix of all training input values:

\[
X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{bmatrix},
\]

Each row of \( X \) contains the \( D \) input parameters for that particular training case run. The training outputs, \( y \), are stacked into a vector of size \( N \times 1 \): \( y = [y_1, y_2, \ldots, y_N]^\top \). Since \( f(x) \) has a GP prior and the likelihood function is also Gaussian, the latent variables can be integrated yielding a Gaussian distribution on the training output [6]:

\[
y \sim \mathcal{N} \left( \mathbf{0}, K(X, X) + \sigma_n^2 \mathbf{I} \right).
\]

In (7), \( K(X, X) \) is the training set covariance matrix and \( \mathbf{I} \) is the identity matrix. The training set covariance matrix is
built by applying the covariance function between each pair of input parameter values [6]:

$$K(X, X) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_N) \\
 k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_N) \\
 \vdots & \vdots & \ddots & \vdots \\
 k(x_N, x_1) & k(x_N, x_2) & \cdots & k(x_N, x_N) \end{bmatrix}. \quad (8)$$

The training set covariance matrix is therefore a full matrix. If the SE covariance function in (3) is used, each diagonal element of $K(X, X)$ is equal to the signal variance, $\sigma_f^2$. Evaluating the covariance function, however, requires the hyperparameter values to be known, which is accomplished by training the GP emulator.

3.2.2. Training. Training or building the emulator consists of learning the hyperparameters that define the covariance and likelihood functions. As discussed earlier, there are two types of hyperparameters in the SE covariance function, the signal variance, $\sigma_f^2$, and the length scales, $l$. The Gaussian likelihood function used presently consists of one hyperparameter, the likelihood noise (variance), $\sigma_n^2$. The complete set of hyperparameters is denoted by $\phi = \{\sigma_f^2, l, \sigma_n^2\}$.

Two ways to learn the hyperparameters will be discussed here: the empirical Bayes approach and the full Bayesian approach. "Full Bayesian" refers to inferring the hyperparameter posterior distribution given the training data. Due to the complexity of the relationship between the hyperparameters and the training output, sampling based Markov Chain Monte Carlo (MCMC) inference is required to perform the full Bayesian approach. The "empirical Bayes" method summarizes the hyperparameters with point estimates. The hyperparameter contribution to the output uncertainty is therefore neglected, but, as discussed by many authors, this is an acceptable approximation [4, 5]. The entire GP model is still considered Bayesian because the GP itself is a statement of the probability of the latent function, which can then make a statement of the probability of the output. The point estimates can be found either from sampling-based approaches or by optimization methods. With optimization procedures, the empirical Bayes approach would be much faster than the full Bayesian training approach. However, cross-validation is very important to ensure the optimizer did not get "stuck" at a local optimum [6].

However, this work used a hybrid approach to training. MCMC sampling was used to draw samples of the hyperparameter posterior distribution just as in the full Bayesian approach. The hyperparameters were then summarized as point estimates at the posterior mean values. Using point estimates greatly reduced the computer memory required to make predictions (which are described later). The sampling based approach removed having to perform cross-validation since the point estimates correspond to the values that on average maximize the posterior density.

The prior distribution on the set of hyperparameters, known as the hyperprior, must be specified as part of the MCMC sampling procedure. The simplest hyperprior would be the “flat” improper hyperprior, $p(\phi) \propto 1$; however for GP models the input and output can be scaled to facilitate meaningful hyperprior specification. Following [2, 3, 5], the inputs used in this work are all scaled between 0 and 1, where 0 and 1 correspond to the minimum and maximum training set value, respectively. Additionally, the training output data are scaled to a standard normal, with mean 0 and variance 1. Since the signal variance, $\sigma_f^2$, defines the diagonal elements of the covariance matrix, it is biased to be near 1. The likelihood noise, $\sigma_n^2$, is biased to be a small value using a Gaussian distribution with prior mean of $10^{-6}$. This hyperprior format biases the sampling procedure to try to find length scale values that match the training output within this noise tolerance. The length scale hyperpriors are more difficult to set, but the formulation from Higdon was used [2, 3, 8], which a priori biases the length scales to yield smooth input/output relationships. Only the training data can reduce the length scales; therefore only the training data can dictate if an input strongly influences the output variability.

Additionally, a small "nugget" or "jitter" term was added to the diagonal elements of $K(X, X)$. The nugget term is rarely mentioned outside of footnotes in most references in the literature [6], but it is a very important part of practical implementations of GP models. The nugget adds a small amount of additional noise, preventing a GP model from interpolating the training set exactly. This additional noise may be very useful at preventing the training set covariance matrix from being ill-conditioned. There have been some detailed investigations into the nugget’s influence on the training algorithm results [5], but for practical purposes the nugget is a simple way to make sure the covariance matrix is always invertible.

The hyperparameter posterior, up to a normalizing constant, can be written as

$$p(\phi | y) \propto p(y | \phi) p(\phi). \quad (9)$$

In (9), $p(\phi)$ is the prior described previously and the likelihood function, $p(y | \phi)$, is (7) rewritten to explicitly depend on the hyperparameters. Hyperparameter posterior samples were drawn using the Adaptive Metropolis (AM) MCMC algorithm [19]. The AM-MCMC algorithm improves the efficiency of the basic Random Walk Metropolis (RWM) sampling algorithm because the MCMC proposal distribution covariance matrix is empirically computed using the previous samples. Regardless of the type of MCMC algorithm used, the likelihood function must be evaluated for each MCMC sample. The log-likelihood, written up to a normalizing constant, is [6]

$$\log \left[ p(y | \phi) \right] \propto -\frac{1}{2} y^T \left[ K(X, X) + \sigma_n^2 I \right]^{-1} y - \frac{1}{2} \log \left[ \left| K(X, X) + \sigma_n^2 I \right| \right]. \quad (10)$$

Equation (10) clearly shows that the training set covariance matrix must be inverted at each MCMC sample. This highlights why the nugget term is useful, if for a particular sample
the likelihood noise, $\sigma_n^2$, does not provide enough noise to allow the matrix to be inverted. The inversion of the training set covariance matrix is the most computationally expensive part of the training algorithm.

3.2.3. Predictions. Once the emulator is trained, predictions can be made at input values that were not part of the training set. If there are $N_e$ new test or prediction points, the test input matrix, $X_e$, is size $N_e \times D$. Under the GP model framework, the latent function at those new test points has the same GP prior as the training points:

$$f_e \sim \mathcal{N}(0, K(X_e, X_e)).$$

Comparing (11) with the training output GP prior in (7), the key difference is that the covariance matrix is evaluated at the test input values rather than the training input values. As written, the test prior provides very little useful information, since it has no information regarding the structure of the training dataset. The test latent output must therefore be conditioned on the training output. The joint prior is a multivariate normal distribution [6]:

$$\begin{bmatrix} y \\ f_e \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_e) \\ K(X_e, X) & K(X_e, X_e) \end{bmatrix} \right),$$

where $K(X, X_e)$ is the cross-covariance matrix between the training and test input values. The cross-covariance matrix is size $N \times N_e$ and $K(X_e, X)$ is its transpose. The posterior predictive covariance of $f_e$ is:

$$\begin{bmatrix} f_e | y \sim \mathcal{N}(\bar{f}_e, \text{cov}(f_e)), \end{bmatrix}$$

with the posterior predictive mean given as

$$\bar{f}_e = \mathbb{E}[f_e | y] = K(X_e, X) \left( K(X, X) + \sigma_n^2 I \right)^{-1} y,$$

and the posterior predictive covariance is

$$\text{cov}(f_e) = K(X_e, X_e) - K(X_e, X) \left( K(X, X) + \sigma_n^2 I \right)^{-1} K(X_e, X).$$

The posterior predictive distribution of the test targets, $y_e$, is the same as the latent posterior predictive distribution except the additional likelihood noise is added:

$$y_e \sim \mathcal{N}(\bar{f}_e, \text{cov}(f_e) + \sigma_n^2 I).$$

Equations (14) and (15) reveal the important features of the GP emulator. First, the posterior predictive covariance shrinks the prior test covariance as witnessed by the subtraction between the first and second terms on the right-hand side of (15). Second, when making predictions at the training points ($X_e = X$), the predictive uncertainty shrinks to the allowable error tolerance.

3.2.4. Gaussian Process-Based Calibration. Once the GP emulator is constructed, it can be used to calibrate the uncertain input parameters, in place of the computer code. Before going into detail of the emulator-based calibration calculations, Bayesian calibration of the computer code itself is reviewed. The computer code functional relationship is denoted as $y(x_{cv}, \theta)$, where $x_{cv}$ is the set of control variables that are not uncertain and $\theta$ are the uncertain input parameters. Control variables are conditioned controlled by experimenters or in a transient could also include time. If the computer code could be used as part of the MCMC sampling, the uncertain parameter posterior distribution (up to a normalizing constant) could be written as

$$p(\theta | y_o) \propto p(y_o | y(x_{cv}, \theta)) p(\theta).$$

In (17), $y_o$ refers to the observational (experimental) data and $x_{cv}$ are the control variables’ locations for the observational data. The computer code therefore acts as a (potentially very nonlinear) mapping function between the uncertain inputs and the observational data. As discussed previously, the computer code is computationally too expensive and the emulator is used in place of the computer code for Bayesian calibration. To facilitate the emulator-based calibration, the likelihood function between the computer prediction and the observational data is split into a hierarchical-like fashion. The total likelihood consists of two parts. The first component is the likelihood between the observational data and the prediction, $p(y_o | y)$. The second part is the likelihood between the computer prediction and the uncertain inputs, $p(y | x_{cv}, \theta)$. The posterior distribution is now the joint posterior distribution between the uncertain inputs and the computer predictions, both conditioned on the observational data:

$$p(y, \theta | y_o) \propto p(y_o | y) p(y | x_{cv}, \theta) p(\theta).$$

The likelihood between the observational data and the computer predictions, $p(y_o | y)$, is the assumed likelihood model for the experiment. This work uses a Gaussian likelihood with known independent measurement errors at each of the observational data points. Assuming $N_o$ independent data points, the likelihood function factorizes as

$$p(y_o | y) = \prod_{i=1}^{N_o} p(y_{oi} | y) = \prod_{i=1}^{N_o} \mathcal{N}(y_i, \sigma_{e,i}^2),$$

where $\sigma_{e,i}^2$ is the measurement error variance for the $i$th observational data point. The likelihood between computer prediction and the inputs, $p(y | x_{cv}, \theta)$, is almost impossible to write analytically because of the very complex nature of the computer code. However, $p(y | x_{cv}, \theta)$ can be approximated using the emulator which leads to the emulator-modified likelihood function. As discussed in Section 2, there are two ways to accomplish this. The alternate “data fusion” approach of [2, 3] uses the GP prior distribution to approximate $p(y | x_{cv}, \theta)$. This work however uses the GP posterior predictive distribution, of the already built emulator, to approximate $p(y | x_{cv}, \theta)$. The training set is denoted as a whole as
\[ \mathcal{D} = \{y, X\}, \] and the hyperparameters are assumed to be already determined as part of the training algorithm. The joint posterior between the emulator estimated predictions \( y_* \) and the uncertain inputs is

\[
p \left( y_* \mid y_o, \mathcal{D}, \phi \right) \propto p \left( y_o \mid y_* \right) p \left( y_* \mid \{x_{v_o}, \theta\}, \mathcal{D}, \phi \right) p \left( \theta \right). \tag{20}
\]

In (20), \( p \left( y_* \mid \{x_{v_o}, \theta\}, \mathcal{D}, \phi \right) \) is exactly the same as (16), except that it is explicitly written to depend on the training set and hyperparameters. Since the GP posterior predictive distribution is Gaussian and the likelihood between the observational data and computer prediction is also Gaussian, the emulator predictions can be integrated out of (20). The observational data and computer prediction is also Gaussian, set and hyperparameters. Since the GP posterior predictive distribution on the uncertain inputs conditioned on the observational data is then

\[
p \left( \theta \mid y_o, \mathcal{D}, \phi \right) \propto p \left( y_o \mid \{x_{v_o}, \theta\}, \mathcal{D}, \phi \right) p \left( \theta \right). \tag{21}
\]

The likelihood between the uncertain inputs and the observational data is the GP emulator-modified likelihood function equal to the GP posterior predictive distribution with the measurement error added to the predictive variance:

\[
y_o \mid \{x_{v_o}, \theta\}, \mathcal{D}, \phi \sim \mathcal{N} \left( \mathbf{f}_*, \text{cov} (\mathbf{f}_*) + \sigma_n^2 I + \Sigma_c \right). \tag{22}
\]

In (22), \( \Sigma_c \) is the measurement error covariance matrix which is assumed to be diagonal. If more complicated likelihood functions between the observational data and computer prediction were assumed, (21) and (22) would potentially be very different and even require approximations. Equation (22) also provides the direct comparison with the “data fusion” approach described in Section 2. The emulator-modified likelihood function given by equation 4 in [2] uses the GP prior mean and covariance matrix, while this work uses the GP posterior predictive mean and covariance matrix.

### 3.3. Function Factorization with Gaussian Process (FFGP) Priors Emulators

For very large datasets, the inverse of the training set covariance matrix might be too expensive to compute. Typically, “very large” corresponds to training sets with over 10,000 points. These situations can occur for several reasons, the obvious being that a large number of computer code evaluations are required. Training sets become very large when the goal is to emulate multiple outputs, especially for time series predictions. If 100 points in time are taken from a single computer code evaluation (referred to as a case run) and 100 cases are required to cover the ranges of the uncertain variables, the total training set consists of 10,000 points.

As stated previously, there are various solutions to this issue, most of which involve some form of a dimensionality reduction technique. The function factorization approach used in this work embeds the dimensionality reduction as part of the emulator through factor analysis techniques. The following sections describe the formulation and implementation of the function factorization model.

#### 3.3.1. Formulation

The main idea of function factorization (FF) is to approximate a complicated function, \( y(x) \), on a high dimensional space, \( \mathcal{X} \), by the sum of products of a number of simpler functions, \( f_{ik}(x_i) \), on lower dimensional subspaces, \( \mathcal{X}^i \). The FF-model is [17]

\[
y(x) \approx \sum_{k=1}^{K} \prod_{i=1}^{I} f_{ik}(x_i). \tag{23}
\]

In (23), \( I \) is the number of different factors and \( K \) is the number of different components within each factor. The function \( f_{ik}(x_i) \) is therefore the latent (hidden) function of the \( k \)th component within the \( i \)th factor. These hidden patterns are not observed directly, but rather must be inferred from the training dataset. The patterns represent a hidden underlying trend within the training data that characterizes the input/output relationship. In the context of emulating safety analysis codes, the patterns correspond to trends between the inputs and the code output of interest, a temperature, for example. With two factors, factor 1 could be the time factor which captures the temperature response through time and factor two could be the trend due to an uncertain input or the interaction of several uncertain inputs. These hidden patterns are not observed directly but interact together to produce the observed temperature response. As will be discussed later, constructing the FF-model requires learning these hidden patterns from the observed training data.

The difference between a factor and component is more distinguishable when (23) is rewritten in matrix form. The training output data will now be denoted as a matrix \( Y \) of size \( M \times N \). In the GP emulator discussion, \( N \) was the number of training points. In the FF-model framework, \( N \) refers to the number of computer code case runs and \( M \) is the number of points taken per case run. If one data point is taken per case run, \( M = 1 \), then the number of case runs equals the number of training points. With two factors, there are two sets of training inputs, \( x_1 \) and \( x_2 \). The inputs do not need to be the same size. If factor 1 corresponds to the number of points taken per case run, then \( x_1 \) is size \( M \times D_1 \). Factor 2 would then correspond to the number of different case runs; thus \( x_2 \) is size \( N \times D_2 \). The entire set of training input values will be denoted as \( X = \{x_1, x_2\} \) and the entire training set will be denoted as for the GP emulator, \( \mathcal{D} = \{X, Y\} \). With 1-component for each factor the FF-model becomes a matrix product of two vectors \( f_1 \) and \( f_2 \):

\[
Y \approx f_1 f_2^T. \tag{24}
\]

For more than one component, each factor is represented as a matrix. The columns within each factor’s matrix correspond to the individual components within that factor. For the 2-factor 2-component FF-model the factor matrices are \( F_1 = [f_{1,1}, f_{1,2}]^T \) and \( F_2 = [f_{2,1}, f_{2,2}]^T \). The FF-model is then [17]

\[
Y \approx F_1 F_2^T. \tag{25}
\]

The elements within each of the factor matrices are the latent variables which represent that factor’s hidden pattern.
and must be learned from the training dataset. Performing Bayesian inference on the FF-model requires specification of a likelihood function between the training output data and the FF-model as well as the prior specification on each factor matrix. In general, any desired likelihood function could be used, but this work focused on a simple Gaussian likelihood with a likelihood noise \( \sigma_f^2 \) and mean equal to the FF-model predictive mean. The likelihood function is therefore the same as the likelihood function between the latent GP variables as the training output, just with the FF-model replacing the GP latent variable. The prior on each component within each factor is specified as a GP prior. Because the FF-model uses a GP, the emulator is known as the FFGP model. As described in detail by Schmidt, this FFGP approach is a generalization of the non-negative matrix factorization (NMF) technique [17]. Each GP prior is assumed to be a zero-mean GP with a SE covariance function, though in general different covariance functions could be used. The GP priors on the \( k \)th component for both the two factors are written as

\[
\begin{align*}
&f_{1,k}(x_1) \sim \mathcal{GP}(0, k_{1,k}(x_1, x_1); \phi_{1,k}) , \\
&f_{2,k}(x_2) \sim \mathcal{GP}(0, k_{2,k}(x_2, x_2); \phi_{2,k}) .
\end{align*}
\] (26)

The semicolon notation within each of the GP priors denotes that both priors depend on the respective set of hyperparameters. Each covariance function consists of a similar set of hyperparameters as those shown in (3), namely, the signal variance and the length scales. An additional nugget hyperparameter, \( \sigma^2_f \), was included to prevent ill-conditioning issues, but rather than fixing its value it was considered a hyperparameter, \( \sigma^2_f \). Writing the GP priors in vector notation requires each of the covariance functions to their respective number of input pairs. Using notation consistent with Section 3.2.1, the GP priors on the \( k \)th component for both factors are

\[
\begin{align*}
&f_{1,k} \sim \mathcal{GP}(0, K_{1,k}(x_1, x_1); \phi_{1,k}) , \\
&f_{2,k} \sim \mathcal{GP}(0, K_{2,k}(x_2, x_2); \phi_{2,k}) .
\end{align*}
\] (27)

Comparing (27) to the GP emulator formulation immediately highlights the key differences between the two emulator types. First, the GP emulator was able to specify a prior distribution on the output data itself, as given by (7), while the FFGP emulator specifies prior distribution on the latent patterns. As described in Section 3.2.1, (7) was actually derived by integrating the GP latent variables. The FFGP latent variables cannot be integrated however, and so the FFGP model requires learning the latent variables as well as the hyperparameters as part of the training algorithm. This adds significant complexity compared to the training of the standard GP emulator. However, this added complexity may enable an important computational benefit. The standard GP emulator covariance matrix consists of the covariance function applied to every input pair in the entire training set. For the present scenario there are a total of \( NM \) training points, which means the covariance matrix is size \( NM \times NM \).

In the FFGP framework, each factor's covariance matrix is constructed by evaluating the factor's covariance function only at each of that particular factor's input pairs. The factor 1 covariance matrix is therefore size \( M \times M \) and the factor 2 covariance matrix is size \( N \times N \). By decomposing the data into various patterns, the FFGP emulator is a dimensionality reduction technique that works with multiple smaller covariance matrices.

3.3.2. Training. Training the FFGP emulator requires learning all of the latent variables and hyperparameters. For notational simplicity, the following set of expressions will assume a 2-factor 1-component FFGP model. The joint posterior for FFGP models with more components is straightforward to write out. Denoting the set of all hyperparameters as \( \Xi = \{ \phi_1, \sigma^2_f \} \), the joint posterior distribution (up to a normalizing constant) between all latent variables and hyperparameters for a 2-factor 1-component FFGP model is

\[
\begin{align*}
&p(f_1, f_2, \Xi | \mathcal{D}) \\
&\propto p(Y | f_1, f_2, \sigma^2_f) p(f_1 | \phi_1) p(f_2 | \phi_2) p(\Xi) .
\end{align*}
\] (28)

The log-likelihood function (up to a normalizing constant) between the training output data and the FF-model is [17]

\[
\begin{align*}
\log p(Y | f_1, f_2, \sigma^2_f) \propto - &\frac{1}{2\sigma_f^2} \left\| Y - f_1 f_2^T \right\|_F^2 \\
&- \frac{NM}{2} \log \sigma^2_f .
\end{align*}
\] (29)

In (29) \( \| \cdot \|_F^2 \) denotes the Frobenius norm. The log prior for each of the factor’s priors is

\[
\begin{align*}
\log p(f_i | \phi_i) \propto &- \frac{1}{2} \log |\mathbf{K}_i| - \frac{1}{2} f_i^T \mathbf{K}_i^{-1} f_i , \quad i = 1, 2.
\end{align*}
\] (30)

The two factors are assumed to be independent \( a\ priori \) in (28). With more components, the setup is the same if all components within each factor are also assumed independent \( a\ priori \). Any correlation between any of the components as well as across the factors is induced by the training data through the likelihood function. Drawing samples from the joint posterior with MCMC does not require any assumptions about the posterior correlation structure. Therefore any data induced posterior correlation can be completely captured by the MCMC inference procedure.

Following Schmidt in [7], the Hamiltonian Monte Carlo (HMC) MCMC scheme was used to build the FFGP emulator. The HMC is a very powerful MCMC algorithm that accounts for gradient information to suppress the randomness of a proposal. See [7, 9, 16] for detailed discussions on HMC. The HMC algorithm is ideal for situations with a very large number of highly correlated variables, as is the case with sampling the latent variables presently.

This work has several key differences from Schmidt’s training algorithm in [17], to simplify the implementation and increase the execution speed. Following [20], the latent variables and hyperparameter sampling were split into a
“Gibbs-like” procedure. A single iteration of the MCMC scheme first samples the latent variables given the hyperparameters and then samples the hyperparameters given the latent variables. The latent variables were sampled with HMC, but the hyperparameters can now be sampled from a simpler MCMC algorithm such as the RWM sampler. Although less efficient compared to the HMC scheme, the RWM performed adequately for this work.

The next key difference relative to Schmidt’s training algorithm was to use an empirical Bayes approach and fix the hyperparameters as point estimates, similar to the hybrid style training algorithm of Section 3.2.2. The hyperparameter point estimates are denoted as \( \hat{\Sigma} \). Once the hyperparameters are fixed, the HMC algorithm is restarted, but now the hyperparameters are considered known.

The end result of the HMC algorithm is a potentially very large number of samples of all of the latent variables. One last simplification relative to Schmidt’s setup was to summarize the latent variable posterior as Gaussians. Their posterior means and covariance matrices were empirically estimated from the posterior samples. All of the latent variables are denoted in stacked vector notation as \( \hat{f} \) and the empirically estimated means of the latent variables are \( \mathbb{E}[\hat{f} | \mathcal{D}, \hat{\Sigma}] \). The empirically estimated covariance matrix of all the latent variables is \( \text{cov}([\hat{f} | \mathcal{D}, \hat{\Sigma}] \). As will be shown in the next section, this assumption greatly simplifying making predictions with the FFGP emulator and ultimately provided a very useful approximation that aided the overall goal of emulator-based Bayesian model calibration.

3.3.3. Predictions. The expressions required to make predictions with the FFGP emulator were summarized briefly in [21], but they will be described in detail here. Prediction with the FFGP emulator consists of two steps: first, make a prediction in the latent factor space and then combine the factor predictions together to make a prediction on the output directly. A latent space posterior prediction is very straightforward following MVN theory and is identical in procedure to posterior predictions with the GP emulator. The joint prior between the training latent variables and test latent variables is written out similar to (12). Before writing the joint prior, the two factors are stacked together into a single stacked vector, \( \tilde{f} = [f_1^T, f_2^T]^T \). Because the factors are independent \textit{a priori}, the stacked covariance matrix is a block diagonal matrix:

\[
\tilde{\Sigma} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}
\]

If more components are used, the individual factor covariance matrices are themselves block diagonal matrices. The training latent variables prior in the stacked notation is

\[
\tilde{f} \sim \mathcal{N}(0, \tilde{\Sigma}).
\]  

The subscript \( f \) is used on the stacked covariance matrix to denote that it is the stacked training covariance matrix. The test latent variables prior in stacked notation is similar to (32):  

\[
\tilde{f}_* \sim \mathcal{N}(0, \tilde{\Sigma}_{*}).
\]  

The subscript \( \ast \) is used on the stacked covariance matrix in (33) to denote that it is the stacked test covariance matrix. The cross-covariance matrix between the training and test points in stacked notation is defined as \( \tilde{\Sigma}_{*} \), which requires evaluating the covariance function between the training and test inputs within each factor. The stacked joint prior is now easily written as

\[
\begin{bmatrix} \tilde{f} \\ \tilde{f}_* \end{bmatrix} \sim \mathcal{N}
\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{K}_f & \tilde{K}_{*f} \\ \tilde{K}_{f*}^T & \tilde{K}_{*} \end{bmatrix} \tilde{\Sigma}^{-1}.
\]

Equation (34) is identical in format to (12), except for one key difference. The joint prior is defined between the training and test latent variables, not between the training output and the test latent variables. Conditioning on the training latent variables, the test latent variable posterior predictive (conditional) distribution is

\[
\tilde{f}_* | \tilde{f}, \hat{\Sigma} \sim \mathcal{N} \left( \mathbb{E}[\hat{f}_* | \hat{f}], \text{cov}(\hat{f}_* | \hat{f}) \right).
\]

The posterior predictive (conditional) mean is

\[
\mathbb{E}[\hat{f}_* | \hat{f}, \hat{\Sigma}] = \tilde{K}_{*f} \mathbb{E}[\hat{f} | \hat{\Sigma}] + \tilde{K}_{*} \mathbb{E}[\hat{\Sigma} | \hat{\Sigma}]
\]

and the posterior predictive (conditional) covariance matrix is

\[
\text{cov}(\hat{f}_* | \hat{f}, \hat{\Sigma}) = \tilde{K}_{*} - \tilde{K}_{*f} \mathbb{E}[\hat{\Sigma} | \hat{\Sigma}] \tilde{K}_{*f}^{-1}. \]

The goal is to make a prediction conditioned on the training dataset, not on particular values of the training latent variables. Therefore the training latent variables must be integrated out using their own posterior distribution computed during the training algorithm. The resulting predictive distribution will be approximated as a Gaussian with the mean estimated using the Law of Total Expectation [7]:

\[
\mathbb{E}[\hat{f}_* | \mathcal{D}, \hat{\Sigma}] = \int \mathbb{E}[\hat{f}_* | \hat{f}, \hat{\Sigma}] p(\hat{f} | \mathcal{D}, \hat{\Sigma}) d\hat{f}.
\]

Substituting in (36) gives

\[
\mathbb{E}[\hat{f}_* | \mathcal{D}, \hat{\Sigma}] = \tilde{K}_{*f} \mathbb{E}[\hat{f} | \mathcal{D}, \hat{\Sigma}] \tilde{K}_{*f}^{-1} \mathbb{E}[\hat{\Sigma} | \mathcal{D}, \hat{\Sigma}].
\]

The expression within the integral of (39) is simply the mean of the (stacked) training latent variables, which was empirically estimated from the posterior MCMC samples from the training algorithm. Thus, the posterior predictive test latent variable means are

\[
\mathbb{E}[\hat{f}_* | \mathcal{D}, \hat{\Sigma}] = \tilde{K}_{*f} \mathbb{E}[\hat{f} | \mathcal{D}, \hat{\Sigma}] \tilde{K}_{*f}^{-1} \mathbb{E}[\hat{\Sigma} | \mathcal{D}, \hat{\Sigma}].
\]
Law of Total Covariance sums up the mean of the predictive conditional covariance with the covariance of the predictive conditions means, which is given as

$$
cov \left( \hat{\mathbf{f}}_s \mid \mathcal{D}, \hat{\mathbf{z}} \right) = \mathbb{E} \left[ cov \left( \hat{\mathbf{f}}_s \mid \hat{\mathbf{f}}, \mathcal{D}, \hat{\mathbf{z}} \right) \right] + cov \left( \mathbb{E} \left[ \hat{\mathbf{f}}_s \mid \hat{\mathbf{f}}, \mathcal{D}, \hat{\mathbf{z}} \right] \right).
$$

(41)

Substituting in (36) and (37) as well as rearranging yields

$$
cov \left( \hat{\mathbf{f}}_s \mid \mathcal{D}, \hat{\mathbf{z}} \right) = \hat{\mathbf{K}}_{ss} - \hat{\mathbf{K}}_{sf}^{-1} \left( \hat{\mathbf{K}}_{sf}^{-1} \mathbb{E} \left[ \hat{\mathbf{f}} \mid \mathcal{D}, \hat{\mathbf{z}} \right] \right) \hat{\mathbf{K}}_{sf}^{-1}.
$$

(42)

Equations (40) and (42) are the approximate posterior predictive test latent variable mean and covariance matrix. They are referred to as being approximate because the training latent variable posterior distribution was approximated as a Gaussian with empirically estimated means and covariance matrix from the training algorithm.

The FF-model predictions can now be estimated. The FF-model predictive distribution is approximated as a Gaussian, with the estimated FF-model predictive means stored in an $M_s \times N_s$ matrix denoted as $\mathbf{H}_s$. $M_s$ is the number of predictive “locations” to be made per case, and $N_s$ is the number of cases to predict. If the FFGP model is emulating a transient, $M_s$ is the number of predictions per case and $N_s$ is the number of prediction cases. In general, the FFGP emulator can therefore make prediction at a large number of points in time (or in general any number of control variable locations, or predictions per case) can be predicted, but for notational convenience it is assumed that the number of predictions per case equals the number of observational locations, $M_s = N_s$. At each MCMC iteration the FFGP emulator predictions are therefore size $N_s \times 1$.

The FF-model approximate predictive mean requires splitting the stacked latent variables into their respective factors:

$$
\hat{\mathbf{f}}_s = \left[ \hat{\mathbf{f}}_{1s}^T, \hat{\mathbf{f}}_{2s}^T \right]^T.
$$

Then the stacked-factor vectors are reshaped into matrices:

$$
\mathbf{F}_{1s} = \text{vec}^{-1} \left( \hat{\mathbf{f}}_{1s} \right),
$$

$$
\mathbf{F}_{2s} = \text{vec}^{-1} \left( \hat{\mathbf{f}}_{2s} \right).
$$

Additionally, the expressions will focus on the predictive FF-model distribution at a single point rather than in vector notation. This simplifies the notation considerably.

The FF-model approximate predictive mean requires computing the expectation of the product of two latent variable factors. At the $(m_s, n_s)$th predictive point the FF-model approximate predictive mean is

$$
\mathbb{E} \left[ \mathbf{H}_s \left( m_s, n_s \right) \right] = \sum_{k=1}^{K} \mathbb{E} \left[ \mathbf{F}_{1s} \left( m_s, k \right) \mathbf{F}_{2s} \left( n_s, k \right) \right].
$$

(45)

The $k$th component in the summation in (45) is the standard result for the product of two correlated random variables:

$$
\mathbb{E} \left[ \mathbf{F}_{1s} \left( m_s, k \right) \mathbf{F}_{2s} \left( n_s, k \right) \right] = \mathbb{E} \left[ \mathbf{F}_{1s} \left( m_s, k \right) \right] \mathbb{E} \left[ \mathbf{F}_{2s} \left( n_s, k \right) \right] + \text{cov} \left( \mathbf{F}_{1s} \left( m_s, k \right), \mathbf{F}_{2s} \left( n_s, k \right) \right).
$$

(46)

The FF-model approximate predictive variance is the variance of the summation of products of random variables plus the FF-model likelihood noise:

$$
\text{var} \left( \mathbf{H}_s \left( m_s, n_s \right) \right) = \sigma_n^2 + \text{var} \left( \sum_{k=1}^{K} \mathbf{F}_{1s} \left( m_s, k \right) \mathbf{F}_{2s} \left( n_s, k \right) \right).
$$

(47)

Writing out the expression completely gives

$$
\text{var} \left( \mathbf{H}_s \left( m_s, n_s \right) \right) = \sigma_n^2 + \sum_{k=1}^{K} \text{var} \left( \mathbf{F}_{1s} \left( m_s, k \right) \mathbf{F}_{2s} \left( n_s, k \right) \right) + 2 \sum_{1 \leq k < k' \leq K} \text{cov} \left( \mathbf{F}_{1s} \left( m_s, k \right) \mathbf{F}_{2s} \left( n_s, k' \right), \mathbf{F}_{1s} \left( m_s, k' \right) \mathbf{F}_{2s} \left( n_s, k \right) \right).
$$

(48)

Both (46) and (48) reveal the FF-model approximate prediction depends on the covariance between all components and all factors. This covariance structure of the posterior test latent variables is induced by the training dataset through the posterior training latent variable covariance structure.

3.3.4. FFGP-Based Calibration. With the FFGP emulator posterior predictive distribution approximated as a Gaussian, a modified likelihood can be formulated in much the same way as the GP emulator-modified likelihood function described in Section 3.2.4. As stated earlier, a single case run is being emulated at each MCMC iteration when calibrating the uncertain inputs; therefore $N_s = 1$. Any number of points in time (or in general any number of control variable locations, or predictions per case) can be predicted, but for notational convenience it is assumed that the number of predictions per case equals the number of observational locations, $M_s = N_s$. At each MCMC iteration the FFGP emulator predictions are therefore size $N_s \times 1$. 

The joint posterior between the FFGP emulator predictions and the uncertain input parameters is

\[
p\left(\mathbf{H}_*, \theta \mid \mathbf{y}_o, \mathcal{D}, \tilde{\Sigma}\right) \propto p\left(\mathbf{y}_o \mid \mathbf{H}_*\right) p\left(\mathbf{H}_* \mid \{x_{cv,o}, \theta\}, \mathcal{D}, \tilde{\Sigma}\right) p(\theta).
\]

The likelihood function between the observational data and the predictions is assumed to be Gaussian with a known observational error matrix, just as in Section 3.2.4. Integrating out the FFGP predictions gives the uncertain input posterior distribution which looks very similar to the expression in (21):

\[
p(\theta \mid \mathbf{y}_o, \mathcal{D}, \tilde{\Sigma}) \propto p(\mathbf{y}_o \mid \{x_{cv,o}, \theta\}, \mathcal{D}, \tilde{\Sigma}) p(\theta).
\]

Assuming the observational error matrix \(\Sigma\) is diagonal, the FFGP modified likelihood function factorizes as

\[
p(\mathbf{y}_o \mid \{x_{cv,o}, \theta\}, \mathcal{D}, \tilde{\Sigma}) = \prod_{l=1}^{N_x} p\left(y_{o,l} \mid \{x_{cv,o,l}, \theta\}, \mathcal{D}, \tilde{\Sigma}\right).
\]

The FFGP-modified likelihood function for each observational data point is then

\[
p\left(y_{o,l} \mid \{x_{cv,o,l}, \theta\}, \mathcal{D}, \tilde{\Sigma}\right) \approx \frac{1}{N_x} \prod_{l=1}^{N_x} \mathcal{N}\left(E\left[\mathbf{H}_* (l)\right], \text{var}\left(\mathbf{H}_* (l)\right) + \sigma^2_e\right).
\]

4. Calibration Demonstration: Friction Factor Model

4.1. Problem Statement. A method of manufactured solutions-type approach is used to verify that the emulator-based calibration process is working as expected. The metric of success is that calibration based on the emulator replicates the calibration results if the computer code itself is used in the same MCMC procedure. The “computer code” in this context is a simple expression that would not actually require an emulator and can therefore be easily used to calibrate any of its inputs. Synthetic “observational” data are generated by setting the uncertain inputs at true values and computing the corresponding output. If the calibration process works as intended, the true values of the uncertain inputs will be learned from the synthetic observational data, within the assumed measurement error tolerance.

A simple friction factor expression is used as the computer code:

\[
f = \exp(b) \, \text{Re}^{-\exp(c)}.
\]

Note that \(f\) in (53) is the friction factor and not related to any of the emulator latent variables. The first demonstration below assumes that only \(b\) is uncertain, while the second demonstration assumes that both \(b\) and \(c\) are uncertain. Note that the friction factor expression in (53) is written in the above form to facilitate specifying Gaussian priors on the uncertain inputs. The typical friction factor expression \((f = B/\text{Re}^C)\) can be recovered by substituting in \(B = \exp(b)\) and \(C = \exp(c)\) into (53). Gaussian priors on \(b\) and \(c\) are therefore equivalent to specifying log-normal priors on \(B\) and \(C\). The prior means on \(b\) and \(c\) equal McAdam’s friction factor correlation values: \(
\log(0.184)\) and \(
\log(0.2)\), respectively. The prior variances on each are set so that 95% of the prior probability covers \(\pm 50\%\) around the prior mean.

Each demonstration follows the emulator-based calibration steps outlined in Figure 1.

4.2. Demonstration for the Case of One Uncertain Parameter. With only \(b\) uncertain, the friction factor expression can be decomposed into the product of two separate functions. The first is a function of the Reynolds number and the second is a function of \(b\):

\[
f = g(b) \, g(\text{Re}),
\]

\[
g(b) = \exp(b),
\]

\[
g(\text{Re}) = \text{Re}^{-\exp(c)}.
\]

The 1-component FFGP emulator should be able to exactly model this expression, within the desired noise level, because the 1-component FFGP emulator is, by assumption, the product of two functions. The control variable is the Reynolds number; therefore the two factors in the FFGP model are the Reynolds number factor (factor 1) and the uncertain input factor (factor 2). The training data was generated assuming 15 case runs, \(N = 15\), and 10 control variable locations per case, \(M = 10\). These numbers were chosen based on the “rule of thumb” for GP emulators that requires at least 10 training points per input [5]. The \(b\) training values were selected at 15 equally spaced points in \(\pm 2\sigma\). The Re training inputs were selected at 10 equally spaced points over an assumed Reynolds number range, between 5000 and 45000. The training data is shown in blue in Figure 2 along with the synthetic observational data in red. The measurement error is assumed to be 10% of the mean value of the friction factor. The Reynolds number is shown in scaled terms where 0 and 1 correspond to the minimum and maximum training value, respectively. Figure 2 clearly shows that the true value of \(b\) falls between two of the training case runs.

Even with only one uncertain parameter, there are actually two inputs to the computer code: \(\text{Re}\) and \(b\). If the standard GP emulator was built, a space filling design would need to be used, such as Latin Hypercube Sampling (LHS), to generate input values that sufficiently cover the input space. The FFGP training set is simpler to generate for this demonstration because each factor’s training input values can be generated independent of the other factor. This illustrates how the FFGP emulator decomposes, or literally factorizes, the training set into simpler, smaller subsets.

The 2-factor 1-component FFGP emulator is built following the training algorithm outlined in Section 3.3.2. The posterior results of the observation space training points are shown in Figure 3. The red dots are the training output
data and although difficult to see, the blue lines are the posterior quantiles on the FFGP training output predictions corresponding to the 5th, 25th, 50th, 75th, and 95th quantiles. The quantiles are tightly packed together representing that the FFGP emulator has very little uncertainty. This meets expectations since by assumption the 2-factor 1-component FFGP emulator should be able to exactly model the product of two functions.

The uncertain \( b \) input is calibrated using the FFGP modified likelihood function. The input is scaled between 0 and 1, which corresponds to a prior scaled mean of 0.5 and prior scaled variance of 0.25. Posterior samples were drawn using the RWM algorithm with the FFGP emulator-modified likelihood function. A total of \( 2 \times 10^4 \) samples were drawn with the first half discarded as burn-in. Figure 4 shows the scaled posterior samples in blue with the true value displayed as the horizontal red line. The mixing rate is very high and the posterior samples are tightly packed around the true value.

Figure 5 shows the estimated posterior distribution in blue relative to the relatively uncertain prior in black. The red line is the true value. Figure 5 illustrates how precise the posterior \( b \) distribution is, confirming that the 2-factor 1-component FFGP emulator is working as expected.

4.3. Demonstration for the Case of Two Uncertain Parameters.

With both \( b \) and \( c \) uncertain, the uncertain input function \( g(b, c) \) cannot be written explicitly. It is expected that the 2-factor 1-component FFGP model will no longer be able to exactly model this relationship since the friction factor is no longer a product of two simple functions. A 3-factor model could be used, but this work focused on 2-factor models for convenience. The 2-factor FFGP model requires additional components to gain the necessary flexibility to handle this. The downside of using only 2-factors requires the uncertain parameter factor (factor 2) to be trained with space filling designs, such as LHS. This work did not focus on finding the absolute “best” training set, which is an active area of research.

The LHS generated training dataset is shown in Figure 6. Fifty case runs were made with 25 points taken per case, \( N = 50 \) and \( M = 25 \). Using more training points helped
guarantee the training dataset would "surround" or cover the observational data. For comparison purposes a standard GP emulator was built for this dataset. The GP emulator training points are shown as circles in Figure 6 and correspond to one point taken per case. The Reynolds numbers selected for the GP emulator training set were chosen as part of the LHS process. The FFGP emulator uses a total of \( NM = 1250 \) training points but the two factor covariance matrices are sizes \((M \times M) = (25 \times 25)\) for factor 1 and \((N \times N) = (50 \times 50)\) for factor 2. The GP emulator covariance matrix is size \((N \times N) = (50 \times 50)\) because only 50 points were used by assumption. If all of the training points were used, the GP emulator covariance matrix would be size \((NM \times NM) = (1250 \times 1250)\). The FFGP emulator setup can therefore drastically reduce the computational burden and facilitate using as many training points as possible.

Examining Figure 6 also shows that both the FFGP and GP emulator training sets are quite poor compared to the training set used in the one uncertain input demonstration. Only a few case runs lie within the error bars of the data and there are very few GP emulator training points near the observational data. It would be relatively easy to keep adding new training points manually for this demonstration to yield a training set that is closer to the observational data. However, in a real problem with many uncertain inputs it may be very difficult to do that manually. This demonstration problem was set up this way to show that the FFGP emulator would outperform the GP emulator due to the pattern recognition capabilities.

A total of 3 emulators were constructed, the standard GP and a 2-factor 1-component and 2-factor 2-component FFGP emulators. Due to different output scaling it is difficult to compare the training results between the GP and FFGP emulators, but a simple approach to compare FFGP performance at the end of the training algorithm is to compare the likelihood noise hyperparameter. The \( \sigma_n^2 \) hyperparameter was reparameterized during the RWM sampling as \( \sigma_n^2 = \exp(2\phi_n) \). The more negative \( \phi_n \) is, the smaller the likelihood noise will be for that particular emulator. Figures 7 and 8 show the sample histories for \( \phi_n \) for the 1-component and 2-component FFGP models, respectively. In each figure, the gray line is the initial guess, the blue line shows the samples, and the red line shows the point estimate. It is very clear that the 2-component FFGP emulator is far more accurate relative to the training set. The \( \phi_n \) point estimate for the 1-component model gives a likelihood standard deviation \( (\sigma_n) \) that is over 45x that of the 2-component emulator. This illustrates the point that the 1-component FFGP emulator is no longer an exact representation of the computer code. The additional component within the 2-component FFGP emulator provides
the extra flexibility needed to match the training set more accurately.

With the FFGP emulators built, they were used to calibrate the uncertain $b$ and $c$ inputs using the FFGP modified likelihood function within the AM-MCMC routine. A total of $10^5$ samples were drawn with the first half discarded as burn-in. The calibrated posterior predictions for the 1- and 2-component FFGP emulators are shown in Figures 9 and 10, respectively. In both figures, the plot on the left shows the posterior calibrated predictions along with all of the training data. The plot on the right zooms in on the posterior calibrated predictions and the observational data. The gray lines are the training data. In both figures, the blue lines are the posterior quantiles (the 5th, 25th, 50th, 75th, and 95th quantiles) of the predictive means and although difficult to see, the black line is the mean of the predictive means. The blue lines therefore represent what the emulator thinks the computer code’s posterior predictive quantiles would be if the computer code had been used. The green band is the total predictive uncertainty band of the emulator, spanning 95% of the emulator prediction probability, and is $\pm 2\sigma$ around the mean of the predictive means. Thus, the green band represents the emulator’s confidence. If the edge of the green band falls directly on top of the outer blue lines, the emulator is essentially perfect and contributes no additional uncertainty to the posterior predictions. A gap between the outer blue lines and the edge of the green band, however, illustrates that the emulator has some associated uncertainty when it makes predictions. The emulator is not perfect, as described in the previous sections, and therefore
some spacing between the green band’s edge and the outer blue lines is expected. However, if the gap width is large, the emulator’s own predictive uncertainty starts to dominate the total predictive uncertainty. Considering these conventions, the 1- and 2-component FFGP emulators can be visually compared quite easily. As shown in Figure 10 the green band is very close to the spread in the blue lines; thus the 2-component FFGP emulator adds very little additional uncertainty in the predictions. The 2-component FFGP emulator’s higher posterior predictive precision relative to the 1-component FFGP emulator is in line with the training results shown in Figures 7 and 8. The 1-component FFGP emulator required more noise to match the training data, which was always propagated through onto the predictions, yielding more uncertain predictions.

Reducing the emulator predictive uncertainty allowed the 2-component FFGP emulator to be more accurate relative to the observational data. As shown in Figure 9, the 1-component FFGP emulator predictions seem to regress the observational data, within the total predictive uncertainty. The 2-component FFGP emulator’s reduced total predictive uncertainty allows the data trend to be captured more accurately.

The $b$ and $c$ inputs were also calibrated using the GP emulator. Once constructed the GP modified likelihood function was used within the AM-MCMC scheme. The same number of samples was drawn as was done for the FFGP case, to provide a direct comparison between the GP-modified and FFGP-modified likelihood functions. The GP-based calibrated posterior predictions are shown in Figure 11 using the same format as the FFGP predictions. The GP emulator adds less uncertainty to the predictions than the 1-component FFGP emulator but is more uncertain and less accurate relative to the data than the 2-component FFGP emulator. The predictions over the first half of the Reynolds numbers are very accurate and are similar to the 2-component FFGP emulator predictions. The latter half of the Reynolds number predictions, however, are worse relative to the 2-component FFGP emulator predictions. The reasons for the difference are best explained by examining the posterior distributions on the $b$ and $c$ parameters.

The posterior distributions from each of the three emulator-based calibration processes are shown in Figures 12, 13, and 14. In all three figures, the black line is the estimated prior, blue is the emulator-based estimated posterior, red is the true value, and green is the estimated posterior when the computer code (the friction factor expression) is used in the AM-MCMC scheme instead of the emulators. Each of the figures is shown over the scaled input ranges, so 0.5 is the scaled prior mean. The computer code-based calibration results find the true values very well, with the posterior
mode lining up nearly exactly with the true values. The posterior variance is limited by the assumed measurement error. Although not shown, the posterior variance decreases as the assumed measurement error is decreased.

Although the GP emulator is capable of finding the correct posterior modes, the (marginal) posterior distributions do not match the computer code-based posterior distributions. Smaller second modes are present in both input parameters. As described in detail in [12], the relatively sparse GP training set is impacting the posterior results. The GP is only able to resolve the overall trend, as illustrated by the GP-based posterior mode roughly corresponding to the computer code-based posterior mode. The posterior tails however cannot be resolved since the emulator’s own predictive variance starts to impact predictions far from the overall trend. The variation in the output data can therefore be explained by the additional noise from the emulator, rather than variation in either of the inputs. The inputs can therefore take on values they would not normally have, since from the emulator’s point of view the prediction overlaps the data's own error. The 1-component FFGP emulator-based results also support this concept, since the posterior distributions in Figure 13 are still quite broad. The emulator is capable of shifting the (marginal) posterior distributions in the correct directions, but the additional emulator uncertainty prevents the MCMC sampling from resolving any additional information about the input values. The 2-component FFGP emulator, however, is so accurate relative to the actual friction factor “computer code,” that its uncertain input (marginal) posterior distributions, as shown in Figure 14, are almost identical to the computer code-based results.

In more complex problems, it is not expected that the FFGP-based results will always be as accurate as in this simple demonstration. However, the FFGP emulator-based calibration is capable of matching the computer code-based calibration results as shown here. In more complex, and realistic situations, the computer code-based results will not be available for comparison, so it was important to verify through this method of manufactured solution problem that the emulator-based process works as expected.

5. Conclusions

The emulator-based calibration approach with the FFGP model was shown above to be capable of reproducing the calibration results obtained when the actual computer code is used in the MCMC sampling. As explored in [11, 17], the efficacy of the FFGP in this application can depend on how the model is structured, but, in cases explored, the additional FFGP emulator was shown to outperform the standard GP emulator, on the given friction factor demonstration problem, because it is capable of efficiently using more training data. This is an important feature because safety analysis problems produce time series predictions which could prove to be computationally expensive for standard GP emulators. Reducing the computational burden would require choosing a limited subset of all of the training runs, which might negatively impact the GP emulator-based results as described in [12]. The friction factor calibration GP-based results presented in this work confirmed those issues. The FFGP emulator however uses pattern recognition techniques to efficiently decompose the training data. The latent or hidden patterns allow more training data to be used which can drastically improve the predictive accuracy of the emulator.

This paper specifically covered the theory and formulation of the FFGP-based calibration approach. Work to presented in a subsequent paper, applies the FFGP-based calibration approach to a more realistic safety analysis scenario, an EBR-II loss of flow transient modeled with RELAP5. As will be shown in that paper, the FFGP-based calibration approach is over 600 times faster than if the RELAP5 model was used directly. Moreover, the FFGP approach is needed, because the standard GP emulator does not provide the
necessary flexibility to emulate the RELAP5 time series predictions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


Research Article

Delayed Station Blackout Event and Nuclear Safety

Andrija Volkanovski and Andrej Prošek

Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia

Correspondence should be addressed to Andrija Volkanovski; andrija.volkanovski@ijs.si

Received 15 January 2015; Accepted 4 March 2015

Academic Editor: Francesco Di Maio

Copyright © 2015 A. Volkanovski and A. Prošek. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The loss of off-site power (LOOP) event occurs when all electrical power to the nuclear power plant from the power grid is lost. Complete failure of both off-site and on-site alternating current (AC) power sources is referred to as a station blackout (SBO). Combined LOOP and SBO events are analyzed in this paper. The analysis is done for different time delays between the LOOP and SBO events. Deterministic safety analysis is utilized for the assessment of the plant parameters for different time delays of the SBO event. Obtained plant parameters are used for the assessment of the probabilities of the functional events in the SBO event tree. The results show that the time delay of the SBO after the LOOP leads to a decrease of the core damage frequency (CDF) from the SBO event tree. The reduction of the CDF depends on the time delay of the SBO after the LOOP event. The results show the importance of the safety systems to operate after the plant shutdown when the decay heat is large. Small changes of the basic events importance measures are identified with the introduction of the delay of the SBO event.

1. Introduction

The main purpose of the nuclear safety is to prevent the release of radioactive materials formed in the fuel and to ensure that the operation of nuclear power plants (NPP) does not contribute significantly to individual and societal health risk [1]. The nuclear safety is assured in all situations with the provision of the basic safety functions: control of reactivity, removal of decay heat to the ultimate heat sink, and confinement of radioactive materials [2]. The systems, structures, and components providing the basic safety functions shall be protected from hazards that may threaten their integrity and intended function. A set of design criteria are defined and required for the protection of the safety functions [3] in all situations.

The currently operating nuclear power plants have reliable active safety systems realizing the basic safety functions. The electrical energy is necessary for powering and controlling the active safety systems. Both off-site and on-site power systems shall be provided, each independent of the other and capable of providing power for all safety functions [4]. The NPP should be designed to sustain a complete loss of off-site power and a single failure within the on-site power system. The complete loss of off-site power is designated as loss of off-site power (LOOP). The NPP has multiple (at least two) redundant sources of alternate electrical power that are normally emergency diesel generators. These sources of alternate electrical power start automatically after LOOP event and deliver power to the corresponding safety systems. The failure of the emergency diesel generators, concurrent with loss of off-site power, is named station blackout event (SBO). During the SBO the batteries with limited capacity provide electrical power to the essential NPP instrumentation and control systems [5]. The station blackout coping time considered in the design is plant specific and depends on multiple factors [6].

The Tohoku-Taiheiyou-Oki Earthquake, which occurred near the east coast of Honshu, Japan [8], resulted in
the power grid failure and consequential LOOP at the six units of the Fukushima Daiichi NPP. The subsequent tsunami caused, due to the flooding, significant damage to the on-site distribution system to at least four of the six units of the Fukushima Daiichi NPP [9]. Following the loss of electric power to normal and emergency core cooling systems and the subsequent failure of back-up decay heat removal systems, water injection into the cores of all three reactors was compromised resulting in core damage [10]. The batteries of Unit 1 and Unit 2 were available for one hour until arrival of tsunami and their submerge [11]. The batteries of Unit 3 were available and provided power for at least 35 hours after the tsunami, resulting in operation of the safety cooling system.

The standard PSA is analysing LOOP and SBO as two separate and independent events. The LOOP event followed by SBO after certain delay is not analysed in the standard PSA. This paper analyses combined LOOP and SBO events for four assumed time delay intervals. The NPP parameters are assessed with deterministic safety analyses.

The description of the NPP model in deterministic safety analyses and developed case scenarios are given in Section 2.1. The description of the PSA model is given in Section 2.2. The implications of the delay on the functional events in the SBO event tree are also discussed in Section 2.2. The main results of the deterministic safety analyses utilized as input to PSA are given in Section 3.1. Obtained PSA results are given in Section 3.2. Main conclusions of the study are presented in Section 4.

2. NPP Models

2.1. Reference Deterministic Model. The RELAP5 input model of the pressurized water reactor (PWR) nuclear power plant is used for the assessment of the nuclear power plant parameters. The RELAP5 input model of an operational two-loop PWR plant is described in detail in studies [5, 12].

The following scenarios with or without reactor coolant pumps (RCP) seal leakage and with or without available turbine driven auxiliary feedwater system (TD AFWS) and pressurizer power operated relief valve (PRZ PORV) stuck open are developed and analysed:

(i) SBOS0-SBO with RCPs, seal loss of coolant accident (LOCA) and TD AFWS operational for 0 h;
(ii) SBONP-SBO without RCPs, seal LOCA and TD AFWS operational and PRZ PORV stuck open after first opening;
(iii) SBOs4-SBO with RCPs, seal LOCA and TD AFWS operational for 4 h.

The modelled NPP has station blackout coping time of 4 hours equal to the TD AFW operational time.

The case scenarios are modified with the consideration of the SBO delay of 15, 30, 60, and 75 minutes following the LOOP. Motor driven AFW pump and instrumentation and control are assumed to be operational for the delay interval in the analysed case scenarios resulting from the availability of alternate current from the emergency diesel generators. Operation of the motor driven AFW pumps results in the injection of water in steam generators after the reactor scram when the decay heat is the largest. This results in the delay of core damage and extension of the available time for restoration of electrical power during station blackout.

The only operator action assumed in the deterministic model is that the steam generator narrow range level is maintained at around 69%.

Obtained results from the analysed scenarios are used as input to the PSA model.

2.2. Reference Probabilistic Model. The reference PSA model is developed on the basis of the Level 1 PSA model of the Surry Unit 1 NPP [13] described in the study [5].

The reference PSA model has 18 event trees, 171 fault trees, and 581 basic events. The CDF is assessed for internal initiating events during the power operation.

The SBO event is modelled in separate event tree given in Figure 1. The SBO event tree, as shown in Figure 1, contains all functional events of a representative SBO event tree for the Westinghouse PWR [5, 14].

The plant coping features successfully mitigate most of such events. Therefore a fraction of the SBO events will lead to the core damage. Feed and bleed is not included in the SBO event tree because pumps available for the feed function require alternate current (AC) power. During the SBO conditions, only the AFW turbine driven pump (TDP) is available for core cooling on the secondary side over the steam generators. Reactor coolant pumps are equipped with staged shaft seals which are provided with cooling system designed to maintain seal integrity. Cooling system is not available during the SBO event resulting in exposure of seals materials to elevated reactor coolant system (RCS) temperatures. Increased temperature result in degradation of the seals materials and increased leakage rate. The sequences in the SBO event tree ending with “CD” contribute to the SBO CDF. The functional events that are affected by the delay of the SBO following the LOOP are marked with red squares in Figure 1.

The SBO event is the first event in the SBO event tree. The frequency of this event is termed the SBO frequency. The SBO frequency is assessed as top event in the SBO fault tree given in Figure 2. The station blackout fault tree, as shown in Figure 2, considers failure of both emergency diesel generators (EDGs) to start and operate, the maintenance unavailability, and failure of the associated circuit breakers. The common-cause failure of both EDGs is considered in addition to the individual EDG failures.

The second functional event NRAC-SGDR in the station blackout event tree corresponds to the restoration of AC power to the plant safety busses before the drying of the coolant on the secondary side of the steam generators (SG). The results of the deterministic analysis for scenarios SBOs0 in Figure 4 show the available time before the SG number 1 is emptied because of drying (i.e., drying time). For SG number 2 the drying times are similar. Figure 4 shows that introduction of the delay between the LOOP and SBO results in a large increase of the SG drying time.

The third functional event RCI-SBO in the SBO event tree corresponds to the preserving of the RCS inventory until AC
Station blackout event tree.

Figure 2: Station blackout fault tree.

power is restored. The failure to reclose the pressurizer power operated relief valves (PORV) results in LOCA.

The fourth functional event SGI-SBO represents the secondary side integrity functional event. In the reference PSA model it is assumed that the steam generator atmospheric relief valves will be inoperable during the station blackout due to the unavailability of control power. The steam relief will be through the safety valves. The safety valves failure to reclose results in uncontrolled depressurization jeopardizing the steam generators integrity.

The fifth functional event AFW-SBO is the top event of a fault tree corresponding to the AFW failure. The input to this
Table 1: Probability of nonrecovery of AC power within given time.

<table>
<thead>
<tr>
<th>Basic event</th>
<th>Restoration time [s]</th>
<th>Probability of nonrecovery of AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NRAC-SGDR [0]</td>
<td>4300</td>
<td>4.74E-01</td>
</tr>
<tr>
<td>NRAC-PRZBV [0]</td>
<td>13620</td>
<td>1.67E-01</td>
</tr>
<tr>
<td>NRAC-OFFSITE</td>
<td>33620</td>
<td>5.46E-02</td>
</tr>
<tr>
<td>NRAC-ONSITE</td>
<td>33620</td>
<td>5.76E-04</td>
</tr>
</tbody>
</table>

Table 2: Initiating events.

<table>
<thead>
<tr>
<th>Initiating event</th>
<th>IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>All loss of coolant accidents</td>
<td>LOCA</td>
</tr>
<tr>
<td>Anticipated transients without scram</td>
<td>ATWS</td>
</tr>
<tr>
<td>Loss of main feedwater</td>
<td>LOFW</td>
</tr>
<tr>
<td>Turbine trip</td>
<td>TUT</td>
</tr>
<tr>
<td>Loss of direct current bus</td>
<td>LODC</td>
</tr>
<tr>
<td>Steam generator tube rupture</td>
<td>SGTR</td>
</tr>
<tr>
<td>Loss of off-site power</td>
<td>LOOP</td>
</tr>
<tr>
<td>Station blackout</td>
<td>SBO</td>
</tr>
</tbody>
</table>

Functional event is the fault tree representing the AFW failure considering multiple failures of the AFW system.

The sixth functional event NRAC-PRZBV corresponds to the restoration of AC power to the plant safety busses and isolation by the pressurizer block valve before start of the core heatup. The available time before the core heatup for this functional event is obtained from scenario SBONP with results given in Figure 6.

The seventh functional event DEP-SBO corresponds to the operator initiated cooling and depressurization of the reactor coolant system during a long-term station blackout.

The eighth functional event SLOCA-NR-ST corresponds to the RCP seal LOCA.

The last two functional events NRAC-OFFSITE and NRAC-ONSITE correspond to the restoration of AC power to the plant safety busses from off-site and on-site power sources before start of the core heatup. The available time for restoration of AC power is assessed from the results of the SBOS4 case scenarios given in Figure 8.

Table 1 shows the probabilities of nonrecovery of AC power within the restoration time in reference PSA model. The available restoration time for analyzed case scenarios is assessed from the results of the deterministic analyses presented in Section 3.1.

Probabilities of nonrecovery of AC power within given time are obtained as probability of exceedance versus duration curve fits of the off-site power to bus recovery times assessed from the statistical data given in [14]. The loss of off-site power initiating event frequency of LOOP = 7.70E-2 events/yr equal to the value in reference model is used [13].

Table 2 shows descriptions of the initiating events in the PSA model with obtained results for the reference PSA model given in Figure 3.

Figure 3 shows that the LOCA is dominant contributor to the plant CDF. The LOOP with share of 21% and the SBO with share of 3% contribute to one-quarter of the overall CDF.

3. Results

3.1. Results of the Deterministic Model. The main parameters characterizing the RELAP5 computer code calculations are given in Figures 4–8. These parameters are the pressurizer pressure, average fuel cladding temperature at the top of the core, and SG number 1 wide range level. The pressurizer pressure is important in order to know when pressurizer relief valves open. The fuel cladding temperature gives information if the core integrity is challenged. Finally, cooling through secondary side could be performed when there is sufficient water inventory (level) in the steam generators.

The available time before the SG drying, related to the second functional event in SBO event tree in Figure 1, is assessed from Figure 4 for SBOS0 case scenario and is given in Table 3.

The pressurizer pressure for SBONP case scenarios is given in Figure 5. The PRZ PORV opens when setpoint of 16.4 MPa is reached and it is assumed that it remains in the stuck open position. This results in the sudden drop of the pressure as shown in Figure 5.

The available time before the core heatup, if PRZ PORV is stuck open after first opening, related to NRAC-PRZBV functional event, is assessed from Figure 6 for SBONP case scenario and is given in Table 4.

The pressurizer pressure for analysed SBOS4 case scenarios is given in Figure 7. The pressurizer safety valve opens when setpoint of 17.2 MPa is reached. This results in the loss of reactor coolant system inventory and start of the core heatup.

Time when core heatup starts resulting from TD AFWS stop, related to the last two events in the SBO event tree, is assessed from Figure 8 for SBOS4 case scenario and is given in Table 5.

3.2. Results of the Probabilistic Model. Table 6 shows the basic events representing nonrecovery of AC power within the restoration time used in the PSA model, given in the second column. The probabilities obtained from the statistical data are given in the third column. Composite probability of exceedance for all LOOP categories [14] is selected as representative data for NRAC-SGDR and NRAC-PRZBV functional event. The grid related LOOP categories that

![Figure 3: CDF for internal initiating events.](image-url)
include switchyard, grid, and weather are selected as relevant for the NRAC-OFFSITE functional event. The plant LOOP category is selected as representative data for the NRAC-ONSITE functional event.

The obtained CDF for the reference PSA model and change of ΔCDF for different time delays are given in Table 7.

Table 7 shows that introduction of the delay of the SBO following the LOOP results in a small decrease of the CDF equal for all assumed time delays. This is result of the small share of the SBO event in the overall CDF of the plant. In the NPP that has larger share of the SBO event in the CDF the obtained ΔCDF will be larger and will depend on the delay of the SBO following the LOOP.

The CDF obtained from the SBO event tree for different delays is given in Table 8, with percentage decrease of the core damage frequency ΔCDF given in Figure 9.

Table 8 and Figure 9 show that 15-minute delay of the SBO, after the LOOP, results in 45% decrease of the SBO CDF. The obtained result is expected considering the importance of the provision of effective core cooling after the reactor trip.

The first 10 basic events identified with largest Fussell-Vesely (FV) importance measure in reference PSA model are given in Table 9. The second column in Table 9 contains basic events with description given in the third column, unavailability in the fourth column, and FV importance measure in the fifth column. The last two columns contain values of risk decrease factor (RDF) and risk increase factor (RIF) importance measures.

Table 9 shows that basic events with largest FV importance measure in reference model, together with the LOOP and SI initiating events (IE), are basic events representing the AFW failure to start and operate.

Table 10 shows basic events with largest FV in the model with the 15-minute delay between the LOOP and the SBO. The same events with identical ordering as in reference model given in Table 9 are identified.

The basic events with the largest FV importance measure in SBO event are given in Table II.

Table II shows that the power restoration events from off-site (power grid) and on-site (emergency diesel generators) electric power sources are most important in the SBO event tree.
The results in the tables show that, in the analyzed PSA model, the introduction of the delay of the SBO following LOOP is not affecting the basic events importance measures.

The results show that the introduction of the time delay results in large decrease of the SBO CDF and corresponding risk. The decay heat is large at the initial period after the reactor shutdown and effective cooling in the initial time period will increase available time before core heatup and core damage.

With different modifications in the NPP the essential safety systems and power supplies can be protected from extreme events that can result in the extended SBO. Those modifications include placement of the safety systems in protected structures or their allocation to the higher elevation.

The coping strategies proposed by the nuclear industry in response to the Fukushima Daiichi accident are based on installed NPP equipment for initial managing of the beyond-design-basis external events [15]. The protection of the installed equipment, as shown by the results of this study, will result in improvement of the plant safety.
Table 7: Obtained CDF and ΔCDF for different time delays.

<table>
<thead>
<tr>
<th>Reference</th>
<th>15 min</th>
<th>30 min</th>
<th>60 min</th>
<th>75 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF [1/yr]</td>
<td>1.55E−05</td>
<td>1.53E−05</td>
<td>1.53E−05</td>
<td>1.53E−05</td>
</tr>
<tr>
<td>ΔCDF [1/yr]</td>
<td>2.00E−07</td>
<td>2.00E−07</td>
<td>2.00E−07</td>
<td>2.00E−07</td>
</tr>
</tbody>
</table>

Table 8: CDF of analyzed case scenarios.

<table>
<thead>
<tr>
<th>Reference</th>
<th>15 min</th>
<th>30 min</th>
<th>60 min</th>
<th>75 min</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF [1/yr]</td>
<td>4.36E−07</td>
<td>2.38E−07</td>
<td>2.26E−07</td>
<td>2.06E−07</td>
</tr>
<tr>
<td>ΔCDF [1/yr]</td>
<td>1.98E−07</td>
<td>2.10E−07</td>
<td>2.30E−07</td>
<td>2.40E−07</td>
</tr>
</tbody>
</table>

Table 9: The basic events with the largest FV importance measure in reference model.

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Description</th>
<th>Nom. value</th>
<th>FV</th>
<th>RDF</th>
<th>RIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOOP</td>
<td>Loss of off-site power IE</td>
<td>7.70E−02</td>
<td>2.45E−01</td>
<td>1.32E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>2</td>
<td>S1</td>
<td>Medium LOCA IE</td>
<td>1.00E−03</td>
<td>2.07E−01</td>
<td>1.26E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>3</td>
<td>AFW-XHE-FO-U1SO</td>
<td>Operator failure to start AFW</td>
<td>8.20E−02</td>
<td>1.40E−01</td>
<td>1.15E+00</td>
<td>2.48E+00</td>
</tr>
<tr>
<td>4</td>
<td>AFW-TDP-FR-2P6HR</td>
<td>AFW failure to run</td>
<td>3.00E−02</td>
<td>1.31E−01</td>
<td>1.15E+00</td>
<td>5.23E+00</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>Manual reactor scram</td>
<td>1.70E−01</td>
<td>1.29E−01</td>
<td>1.15E+00</td>
<td>1.63E+00</td>
</tr>
<tr>
<td>6</td>
<td>ATWS</td>
<td>Anticipated transient WS IE</td>
<td>5.00E−04</td>
<td>1.29E−01</td>
<td>1.15E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>7</td>
<td>HPI-XHE-FO-FDBLD</td>
<td>Operator failure feed/bleed</td>
<td>7.10E−02</td>
<td>1.28E−01</td>
<td>1.15E+00</td>
<td>2.68E+00</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
<td>Interfacing LOCA IE</td>
<td>1.60E−06</td>
<td>1.03E−01</td>
<td>1.12E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>9</td>
<td>T2</td>
<td>Loss of main FW IE</td>
<td>9.40E−01</td>
<td>9.44E−02</td>
<td>1.10E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>10</td>
<td>RECOV-S1-2</td>
<td>Operator recovery action</td>
<td>9.29E−01</td>
<td>8.40E−02</td>
<td>1.09E+00</td>
<td>1.01E+00</td>
</tr>
</tbody>
</table>

Table 10: Basic events with the largest FV importance measure, model with 15-minute delay.

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Description</th>
<th>Nom. value</th>
<th>FV</th>
<th>RDF</th>
<th>RIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOOP</td>
<td>Loss of off-site power IE</td>
<td>7.70E−02</td>
<td>2.35E−01</td>
<td>1.31E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>2</td>
<td>S1</td>
<td>Medium LOCA IE</td>
<td>1.00E−03</td>
<td>2.10E−01</td>
<td>1.27E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>3</td>
<td>AFW-XHE-FO-U1SO</td>
<td>Operator failure to start AFW</td>
<td>8.20E−02</td>
<td>1.40E−01</td>
<td>1.15E+00</td>
<td>2.49E+00</td>
</tr>
<tr>
<td>4</td>
<td>AFW-TDP-FR-2P6HR</td>
<td>AFW failure to run</td>
<td>3.00E−02</td>
<td>1.32E−01</td>
<td>1.15E+00</td>
<td>5.25E+00</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>Manual reactor scram</td>
<td>1.70E−01</td>
<td>1.31E−01</td>
<td>1.15E+00</td>
<td>1.64E+00</td>
</tr>
<tr>
<td>6</td>
<td>ATWS</td>
<td>Anticipated transient WS IE</td>
<td>5.00E−04</td>
<td>1.31E−01</td>
<td>1.15E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>7</td>
<td>HPI-XHE-FO-FDBLD</td>
<td>Operator failure feed/bleed</td>
<td>7.10E−02</td>
<td>1.30E−01</td>
<td>1.15E+00</td>
<td>2.70E+00</td>
</tr>
<tr>
<td>8</td>
<td>V</td>
<td>Interfacing LOCA IE</td>
<td>1.60E−06</td>
<td>1.05E−01</td>
<td>1.12E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>9</td>
<td>T2</td>
<td>Loss of main FW IE</td>
<td>9.40E−01</td>
<td>9.56E−02</td>
<td>1.11E+00</td>
<td>Infinite</td>
</tr>
<tr>
<td>10</td>
<td>RECOV-S1-2</td>
<td>Operator recovery action</td>
<td>9.29E−01</td>
<td>8.51E−02</td>
<td>1.09E+00</td>
<td>1.01E+00</td>
</tr>
</tbody>
</table>

Table 11: Basic events with the largest FV importance measure in SBO event tree.

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Description</th>
<th>Nom. value</th>
<th>FV</th>
<th>RDF</th>
<th>RIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOOP</td>
<td>Loss of off-site power IE</td>
<td>7.70E−02</td>
<td>1.00E+00</td>
<td>Infinite</td>
<td>Infinite</td>
</tr>
<tr>
<td>2</td>
<td>NRAC-SGDR</td>
<td>Off-site power restoration</td>
<td>4.74E−01</td>
<td>1.00E+00</td>
<td>Infinite</td>
<td>2.11E+00</td>
</tr>
<tr>
<td>3</td>
<td>SBO-PORV-DMD</td>
<td>Primary system integrity</td>
<td>4.50E−01</td>
<td>7.02E−01</td>
<td>3.36E+00</td>
<td>1.86E+00</td>
</tr>
<tr>
<td>4</td>
<td>NRAC-PRZBV</td>
<td>Off-site power restoration</td>
<td>1.67E−01</td>
<td>6.76E−01</td>
<td>3.09E+00</td>
<td>4.37E+00</td>
</tr>
<tr>
<td>5</td>
<td>RECOV-T1SN-21</td>
<td>Operator recovery action</td>
<td>9.33E−01</td>
<td>6.71E−01</td>
<td>3.04E+00</td>
<td>1.05E+00</td>
</tr>
<tr>
<td>6</td>
<td>BETA-2DG</td>
<td>EDG common-cause failure</td>
<td>3.80E−02</td>
<td>4.38E−01</td>
<td>1.78E+00</td>
<td>1.21E+01</td>
</tr>
<tr>
<td>7</td>
<td>OEP-DGN-FS</td>
<td>EDG fails to start</td>
<td>2.20E−02</td>
<td>4.38E−01</td>
<td>1.78E+00</td>
<td>2.05E+01</td>
</tr>
<tr>
<td>8</td>
<td>OEP-DGN-FS-DGO1</td>
<td>EDG #1 fails to start</td>
<td>2.20E−02</td>
<td>3.78E−01</td>
<td>1.60E+00</td>
<td>1.76E+01</td>
</tr>
<tr>
<td>9</td>
<td>OEP-DGN-FS-DGO4</td>
<td>EDG #2 fails to start</td>
<td>2.20E−02</td>
<td>3.78E−01</td>
<td>1.60E+00</td>
<td>1.76E+01</td>
</tr>
<tr>
<td>10</td>
<td>PPS-SOV-O0-1456</td>
<td>RCS PORV fails to reclose</td>
<td>3.00E−02</td>
<td>3.51E−01</td>
<td>1.54E+00</td>
<td>1.24E+01</td>
</tr>
<tr>
<td>11</td>
<td>PPS-SOV-O0-1455</td>
<td>RCS PORV fails to reclose</td>
<td>3.00E−02</td>
<td>3.51E−01</td>
<td>1.54E+00</td>
<td>1.24E+01</td>
</tr>
</tbody>
</table>
Table 12: The basic events with largest FV importance measure in SBO event tree, model with 15-minute delay.

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Description</th>
<th>Nom. value</th>
<th>FV</th>
<th>RDF</th>
<th>RIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOOP</td>
<td>Loss of off-site power IE</td>
<td>7,70E – 02</td>
<td>1,00E + 00</td>
<td>Infinite</td>
<td>Infinite</td>
</tr>
<tr>
<td>2</td>
<td>NRAC-SGDR (15)</td>
<td>Off-site power restoration</td>
<td>2,65E – 01</td>
<td>1,00E + 00</td>
<td>Infinite</td>
<td>3,77E + 00</td>
</tr>
<tr>
<td>3</td>
<td>SBO-PORV-DMD</td>
<td>Primary system integrity</td>
<td>4,50E – 01</td>
<td>6,98E – 01</td>
<td>3,31E + 00</td>
<td>1,85E + 00</td>
</tr>
<tr>
<td>4</td>
<td>NRAC-PRZBV (15)</td>
<td>Off-site power restoration</td>
<td>1,62E – 01</td>
<td>6,71E – 01</td>
<td>3,04E + 00</td>
<td>4,47E + 00</td>
</tr>
<tr>
<td>5</td>
<td>RECOV-TIEN-1</td>
<td>Operator recovery action</td>
<td>9,33E – 01</td>
<td>6,66E – 01</td>
<td>2,99E + 00</td>
<td>1,05E + 00</td>
</tr>
<tr>
<td>6</td>
<td>BETA-2DG</td>
<td>EDG common-cause failure</td>
<td>3,80E – 02</td>
<td>4,38E – 01</td>
<td>1,78E + 00</td>
<td>2,12E + 01</td>
</tr>
<tr>
<td>7</td>
<td>OEP-DGN-FS</td>
<td>EDG fails to start</td>
<td>2,20E – 02</td>
<td>4,38E – 01</td>
<td>1,78E + 00</td>
<td>2,05E + 01</td>
</tr>
<tr>
<td>8</td>
<td>OEP-DGN-FS-DGO4</td>
<td>EDG #1 fails to start</td>
<td>2,20E – 02</td>
<td>3,78E – 01</td>
<td>1,60E + 00</td>
<td>1,76E + 01</td>
</tr>
<tr>
<td>9</td>
<td>OEP-DGN-FS-DGO1</td>
<td>EDG #2 fails to start</td>
<td>2,20E – 02</td>
<td>3,78E – 01</td>
<td>1,60E + 00</td>
<td>1,76E + 01</td>
</tr>
<tr>
<td>10</td>
<td>PPS-SOV-OO-1455</td>
<td>RCS PORV fails to reclose</td>
<td>3,00E – 02</td>
<td>3,49E – 01</td>
<td>1,54E + 00</td>
<td>1,23E + 01</td>
</tr>
<tr>
<td>11</td>
<td>PPS-SOV-OO-1456</td>
<td>RCS PORV fails to reclose</td>
<td>3,00E – 02</td>
<td>3,49E – 01</td>
<td>1,54E + 00</td>
<td>1,23E + 01</td>
</tr>
</tbody>
</table>

4. Conclusion

In this paper the consequences of the delay of the SBO event following the LOOP event are analysed and the results presented. The analysis is done with standard deterministic and probabilistic safety analysis methods and tools for different assumed time delays between the LOOP and the SBO event. The available restoration times are assessed from the deterministic safety analyses results. The probability of nonrecovery of alternate electric power in the restoration times is assessed from the statistical data and inserted in the station blackout event tree.

The results show that small decrease of the overall CDF is obtained with the introduction of the SBO delay following the LOOP. This is because of small share of the SBO event in the overall CDF of the plant. Large decrease of the SBO CDF is obtained with the introduction of the delay. Increase of the delay results in the decrease of the SBO CDF. Introduction of the delay in the analysed model results in small changes of the importance measures.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was partly supported by the Slovenian Research Agency (Research Program P2-0026) and partly by the European Atomic Energy Community’s (Euratom) Seventh Framework Programme FP7/2007–2011 under Grant Agreement no. 605001.

References


Risk-Based Clustering for Near Misses Identification in Integrated Deterministic and Probabilistic Safety Analysis

Francesco Di Maio, Matteo Vagnoli, and Enrico Zio

1Department of Energy, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy
2Chair on System Science and Energetic Challenge, Fondation EDF-Electricite de France, Ecole Centrale Paris and Supelec, Chatenay-Malabry Cedex, 92295 Paris, France

Correspondence should be addressed to Francesco Di Maio; francesco.dimaio@polimi.it

Received 16 December 2014; Accepted 16 February 2015

In integrated deterministic and probabilistic safety analysis (IDPSA), safe scenarios and prime implicants (PIs) are generated by simulation. In this paper, we propose a novel postprocessing method, which resorts to a risk-based clustering method for identifying Near Misses among the safe scenarios. This is important because the possibility of recovering these combinations of failures within a tolerable grace time allows avoiding deviations to accident and, thus, reducing the downtime (and the risk) of the system. The postprocessing risk-significant features for the clustering are extracted from the following: (i) the probability of a scenario to develop into an accidental scenario, (ii) the severity of the consequences that the developing scenario would cause to the system, and (iii) the combination of (i) and (ii) into the overall risk of the developing scenario. The optimal selection of the extracted features is done by a wrapper approach, whereby a modified binary differential evolution (MBDE) embeds a K-means clustering algorithm.

The characteristics of the Near Misses scenarios are identified solving a multiobjective optimization problem, using the Hamming distance as a measure of similarity. The feasibility of the analysis is shown with respect to fault scenarios in a dynamic steam generator (SG) of a nuclear power plant (NPP).

1. Introduction

Integrated deterministic and probabilistic safety analysis (IDPSA) attempts to overcome some limitations of deterministic safety analysis (DSA) and probabilistic safety analysis (PSA). The former is solidly founded on the multibarrier and defense-in-depth concepts and aims at verifying the capability of a nuclear power plant (NPP) to withstand a set of postulated design basis accidents (DBA) [1, 2]. To account for the uncertainties in the model representation of the actual plant behavior, conservatism is introduced in the calculations by thermal-hydraulics (TH) codes under DBA conditions [3]. The latter aims at considering a wider set of possible accidental scenarios and includes the quantification of accident probabilities [4, 5].

Both DSA and PSA are scenario-based analyses, where scenario selection and definition are done by expert judgment. State of the art of DSA and PSA approaches can provide relevant and important insights into what is already known to be an “issue,” but they are not capable of revealing what, and to what extent, is not known (i.e., scenarios which are not expert-selected in the DSA and PSA inputs), with the risk of neglecting or underestimating potentially dangerous scenarios [6]. This is due to the difficulties of the static structure of the classic DSA and PSA approaches in treating dynamic variations that usually occur during the operational time of a process [7] due to (i) stochastic disturbances (e.g., equipment failures), (ii) deterministic plant responses (i.e., transients), (iii) controls, and (iv) operator actions [6, 8, 9]. Indeed, the order and timing of the events occurring along a scenario and the values of the process variables at the time of event occurrence are critical in determining the evolution of the scenario itself [10].

The development and application of IDPSA in practice must meet the challenge of computational complexity, in both model construction and implementation and in
postprocessing for the retrieval of the relevant information from the scenario outcomes. The number of dynamic scenario branches generated in IDPSA increases in power law with the number of occurring events and, thus, is much larger than in classical PSA based on event trees (ET) and fault trees (FT). The a posteriori information retrieval (postprocessing) then becomes quite burdensome and difficult [11, 12]. Continuous event trees (CETs) [13, 14] and dynamic event trees (DET) [15, 16] provide realistic frameworks for IDPSA. However, their application is limited by their computationally intensive nature, by the need of tailoring the algorithms to the system under consideration and by the need of processing a massive amount of data for any single initiating event considered [17].

Postprocessing, in general, consists in classifying the generated dynamic scenarios into safe scenarios and prime impliants (PIs), that is, sequences of events that represent minimal combinations of accident failures necessary for system failure and cannot be covered by more general impliants [18]. Among the safe scenarios, Near Misses are important scenarios to be identified, because they are those sequences of events that reach values of the safety parameters close to, but not exceeding, the corresponding acceptable thresholds [19]. They can, thus, be relevant contributors to the “hidden” risk of the system and should not be neglected, as a small deviation may transform them into accidental scenarios.

In the literature, several authors introduce the concept of Near Misses as accident precursors [20, 21]. We here consider Near Misses as sequences of events that incidentally keep the system in a safe state but endangered and insecure. For the purpose of the analysis, they are here defined as sequences of events similar to those leading the system into fault conditions, except for one characteristic which is missing or is slightly different (e.g., sequence time lag, different failure magnitude, and different involved component in an event) [22].

The postprocessing analysis entails a “Forward” classification of the dynamic scenarios into classes, that is, safe, PIs, and Near Misses and a “Backward” identification of the similarities of the features of the scenarios (i.e., stochastic event occurrence and deterministic process variables values), which characterize the groups of Near Misses among the whole set of safe scenarios.

For the “Forward” classification of the Near Misses sequences, we look at two factors of risk: the probability of occurrence of an undesired event and the severity of the consequence caused by the event [23]. Thus, we describe the sequences of events by (i) the probability \( p \) that the developing scenario is an accidental scenario, (ii) the consequence \( c \) that the developing scenario can cause to the system, and (iii) the overall risk \( r \) of the developing scenario that we compute synthetically as \( r = p \times c \) (expected consequence).

The optimal features for discerning the Near Misses from the safe scenarios are extracted from the profiles of \( p, c, \) and \( r \) of the accidental scenarios and selected by a wrapper algorithm, which takes into account six statistical indicators of \( p, c, \) and \( r \), and, through a modified binary differential evolution (MBDE) optimization algorithm, selects the best features, which are fed to a \( K \)-means clustering algorithm, which is a simple and well-known clustering algorithm (other classical clustering algorithms, such as mean-shift [24, 25] or fuzzy C-means [19, 26]).

The outcomes of this “Forward” classification is, then, interpreted by a “Backward” identification of the similarities of the features of the Near Misses scenarios: the acquired knowledge can be exploited in an online integrated risk monitoring system that can rapidly detect the problem and set up a repair strategy of the occurring failures before the system reaches a fault state.

The proposed approach is illustrated with reference to scenarios occurring in the steam generator (SG) of a NPP [27]. We use multiple-valued logic (MVL) theory for modeling the behavior of the system, where timing and sequences of component failure events are determining the system behavior [4]. By using MVL, we increase the limited description capability of binary variables in modeling the different component operational states (e.g., a valve that can be closed, partially closed, or fully open or can fail at different times) and, therefore, perform an IDPSA postprocessing analysis on the whole set of simulated accidental scenarios [17].

The paper is organized as follows. In Section 2, the SG model used to generate the scenarios for the reliability analysis is presented [27], along with multistate representation of the system dynamics. In Section 3, the PIs are identified and the risk-based “Forward” and “Backward” Near Misses identification method is introduced with reference to the case study considered. In Section 4, conclusions and remarks are drawn.

2. Case Study

2.1. The U-Tube Steam Generator (UTSG) Model. The U-tube steam generator (UTSG) under consideration is sketched in Figure 1. The improper control of the water level, whose difficulties arise from nonminimum phase plant characteristics, that is, plant strong inverse response behavior, particularly at low operating power, due to the so-called “swell and shrink” effects [28], is a major cause of NPP unavailability [28–30].

The reactor coolant enters the UTSG at the bottom and moves upward and then downward in the inverted U-tubes, transferring heat to the secondary fluid before exiting at the bottom. The secondary fluid, the feedwater \((Q_f)\), enters the UTSG at the top of the downcomer, through the space between the tube bundle wrapper and the SG shell. The value of \(Q_f\) is regulated by a system of valves: a low flow rate valve, used when the operating power \((P_o)\) is smaller than 15% of nominal power \((P_{n})\), and a high flow rate valve when \(P_o > 0.15P_n\) [27]. In the secondary side of the tube bundle, water heats up, reaches saturation, starts boiling, and turns into a two-phase mixture. The two-phase fluid moves up through the separator/riser section, where steam is separated from liquid water, and through the dryers, which ensure that the exiting steam \((Q_s)\) is essentially dry. The separated water is recirculated back to the downcomer. The balance between the exiting \(Q_s\) and the incoming \(Q_f\) governs the change in the water level in the SG. Because of the two-phase nature, two types of water level measurements are considered, as shown in Figure 1, each reflecting a different level concept: the narrow
range level \( N_{rl} \) is calculated by pressure difference between two points close to the water level and indicates the mixture level, whereas the wide range level \( W_{rl} \) is calculated by pressure difference between the two extremities of the SG (steam dome and bottom of the downcomer) and indicates the collapsed liquid level that is related to the mass of water in the SG.

“Swell and shrink” phenomena are also modeled to reproduce the dynamic behavior of the SG: when \( Q_e \) increases, the steam pressure in the steam dome decreases and the two-phase fluid in the tube bundle expands causing \( N_{rl} \) to initially swell (i.e., rise), instead of decreasing as would have been expected by the mass balance; contrarily, if \( Q_e \) decreases or \( Q_s \) increases, a shrink effect occurs. A similar model has been presented in [27].

The \( N_{rl} \) is governed by \( Q_e \) and \( Q_s \) across the tube bundle region of the SG as shown by the following transfer function:

\[
N_{rl}(s) = \frac{1}{T_n s} \left( Q_{ef}(s) - Q_{CV}(s) \right),
\]

where \( Q_{ef} \) is the flow rate of the incoming water in the tube bundle, \( Q_{CV} \) is the equivalent steam-water mixture flow rate exiting the tube bundle region, \( T_n \) is a time constant that accounts for the \( N_{rl} \) dynamics.

The incoming water flow rate \( Q_{ef} \) is proportional to \( Q_e \):

\[
Q_{ef}(s) = \frac{1}{(1 + T_h s)(1 + \tau s)} Q_e(s),
\]

where the lag \( 1/(1 + \tau s) \) accounts for the feed-water valve dynamics and \( 1/(1 + T_h s) \) accounts for the water mass transportation dynamics: their values are reported in Table 1.

The exiting steam-water mass \( Q_{CV} \) is proportional to \( Q_s \):

\[
Q_{CV}(s) = \frac{1 - F_g T_g s^2}{1 + T_g s^2} Q_s(s),
\]

where the first-order lag \( 1/(1 + T_g s) \) accounts for the elapsed time from the turbine steam demand and the increase of \( Q_{CV} \), and the nonminimum phase term \( (1 - F_g T_g s)/(1 + T_g s) \) accounts for the two-phase swell and shrink effects.

Combining (1), (2), and (3), \( N_{rl} \) is equal to

\[
N_{rl}(s) = \frac{1}{T_n s} \left( Q_e(s) - Q_e(s) - Q_s(s) \right),
\]

and \( W_{rl} \), that is, the overall water mass in the steam generator, is

\[
W_{rl}(s) = \frac{1}{T_{int} s} (Q_e(s) - Q_s(s)),
\]

where \( T_{int} \) is a time constant that accounts for the \( W_{rl} \) dynamics.

We assume \( y_1 = N_{rl} \) and \( y_2 = W_{rl} \) and \( u = Q_e \) and \( d = Q_s \); the state space representation of the SG model is thus,

\[
\dot{x}(t) = \begin{pmatrix} 0 & 0 & 0 & \frac{1}{T_n} \\ -\frac{1}{T_h} & 0 & -\frac{1}{T_n} & 0 \\ 0 & -\frac{1}{T_g} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix} u(t) + \begin{pmatrix} -\frac{1}{T_n} \\ 0 \\ \frac{1 + F_g T_g}{T_n} \\ 0 \end{pmatrix} d(t),
\]

\[
y(t) = \begin{pmatrix} T_n & 0 & 0 & \tau T_{int} \end{pmatrix} x(t).
\]

The values of the parameters \( T_h, T_n, F_g, \tau, T_g, \) and \( T_{int} \) change depending on the power \( P_o \), as shown in Table 1.

The goal of the system is to maintain the SG water level at a reference position \( N_{ref} \): the SG fails if the \( N_{rl} \) rises (falls) above (below) the threshold \( N_{high} \) \( N_{low} \), in which case automatic reactor or turbine trips are triggered. Indeed, if the \( N_{rl} \) exceeds \( N_{high} \), the steam separator and dryer lose their functionality and excessive moisture is carried in \( Q_s \), degrading the turbine blades profile and the turbine efficiency; if \( N_{rl} \) decreases below \( N_{low} \), insufficient cooling capability of the primary fluid occurs. Similarly, the \( W_{rl} \) is relevant for the cooling capability of the primary circuit [28]. Prealarms are triggered when \( N_{rl} \) exceeds \( N_{rl,ref} \) (\( N_{rl} \)) if a small deviation from \( N_{ref} \) occurs or when \( N_{rl} \) exceeds \( N_{rl,th} \) (\( N_{rl} \)), when the deviation is large. Set points of \( N_{ref} \) and of \( N_{rl} \) depend on \( P_o \), as shown in Figure 2, and, thus, also the alarms thresholds depend on \( P_o \). The \( N_{rl} \) set point is low at low \( P_o \), to
Table 1: Parameters of the UTSG model at different power levels [27].

<table>
<thead>
<tr>
<th>$P_o$</th>
<th>0.03 × $P_n$</th>
<th>0.04 × $P_n$</th>
<th>0.09 × $P_n$</th>
<th>0.24 × $P_n$</th>
<th>0.30 × $P_n$</th>
<th>0.50 × $P_n$</th>
<th>$P_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_n$</td>
<td>36</td>
<td>56</td>
<td>63</td>
<td>44</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>$F_g$</td>
<td>13</td>
<td>18</td>
<td>10</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$T_h$</td>
<td>170</td>
<td>56</td>
<td>30</td>
<td>10</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>$\tau$</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>$T_g$</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
</tr>
<tr>
<td>$T_{int}$</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
<td>140</td>
</tr>
</tbody>
</table>

Figure 2: Set point for $N_{rl}$ at different power rate $P_o$ values.

partially account for the strong inverse response of $N_{rl}$ [28]; thus, the low level thresholds are more restrictive than the high level thresholds at low $P_o$.

A dedicated model has been implemented in SIMULINK to simulate the dynamic response of the UTSG at different $P_o$ values. Both feedforward and feedback digital control schemes have been adopted. The feedback controller is a PID that provides a flow rate $Q_{pid}$ resulting from the residuals between $N_{rl}$ and $N_{ref}$, whereas the feedforward controller operates a safety relief valve that is opened if and only if $N_{rl}$ exceeds the $N_{hl}$ and removes a constant flow safety flow rate ($Q_{sf}$). The block diagram representing the SIMULINK model of the SG is shown in Figure 3: the controlled variable is $N_{rl}$, whereas the control variable is $Q_o$.
2.2. The Set of Possible Failures. The set of multiple component failures that can occur during the system life are shown in Figure 4.

(1) The outlet steam valve can fail stuck at a random time in $[0, 4000]$ (s) in three different positions: (i) closed; (ii) stuck open at 50% of the nominal $Q_o$ that should be provided at $P_o$; (iii) stuck open at 150% of the nominal $Q_o$ that should be provided at $P_o$.

(2) The safety relief valve can fail stuck at a random time in $[0, 4000]$ (s) at a uniform random value $Q_{rl}$ in the range $[0.5, 50.5]$ (kg/s).

(3) The communication between the sensor that monitors $N_{sf}$ and the PID controller can fail at random times in $[0, 4000]$ (s), in which case the PID is provided with the same input value of the previous time step.

(4) The PID controller can fail stuck at random times in $[0, 4000]$ (s), providing a uniform random flow rate $Q_{pid}$ belonging to $[-18, 18]$% of the nominal $Q_e$ that should be provided at $P_o$.

It is worth noticing that in the UTSG there are two PID controllers and, thus, two communications between the sensors measuring $N_{sf}$ and the PIDs (one for high power feedback control and the other for low power feedback control). The selective action of the PIDs depending on $P_o$ hides some of the failures. For example, if the power profile of the scenario under investigation is a ramp, both PIDs are called in operation: if anyone (or both) failed, their fault state is detectable. On the contrary, if we consider scenarios with constant power profile, for example, low power rate ($P_o < 15\% P_{nom}$), the occurrence of a high power feedback control failure cannot be detected, and, thus, the fault remains hidden.

Choices and hypotheses for modeling the failures (i.e., the mission time, the number and type of faults, the distributions of failure times, and magnitudes) have been arbitrarily made with the aim of generating multiple failures in the sequences and capturing the dynamic influence of their order, timing, and magnitude. The choice of a mission time ($T_{miss}$) equal to 4000 (s) has been made, because it is a long enough interval of time to allow the complete development also of slow dynamic accident scenarios.

2.3. The Multistate Representation of System Dynamics. For realistically treating the dynamic behavior of the UTSG when component failures occur, we go beyond the binary state representation and adopt a multiple value logic (MVL) [17, 32] for an approximated description of the continuous time of occurrence of component failures and their magnitude. The MVL allows describing that the components can fail at any (discrete) time (not only the initial time) along the scenario, with different (discrete) magnitudes (not only the most conservative). The discretization of the time and magnitudes values is as follows:

(i) time discretization: we use the labels $t_{mvl} = 1$, $t_{mvl} = 2$, $t_{mvl} = 3$, and $t_{mvl} = 4$, for failures occurring in the intervals $[0, 1000]$ (s), $[1001, 2000]$ (s), $[2001, 3000]$ (s), and $[3001, 4000]$ (s), respectively; if the label $t_{mvl} = 0$, the component does not fail within the time of the whole scenario, $T_{miss}$;

(ii) magnitude discretization:

(a) the steam valve magnitude is indicated as 1, 2, or 3 for failure states corresponding to stuck at 0%, stuck at 50%, and stuck at 150% of the $Q_e$ value that should be provided at $P_o$, respectively; if the steam valve magnitude is indicated as 0, the component does not fail in $T_{miss}$;

(b) the safety relief valve fails with magnitude indicated as 1, 2, 3, and 4, if it is stuck between $[0.5, 12.6]$ (kg/s), $[12.6, 25.27]$ (kg/s), $[25.27, 37.91]$ (kg/s), and $[37.91, 50.5]$ (kg/s), respectively; if the safety relief valve magnitude is indicated as 0, the component does not fail in $T_{miss}$;

(c) the communication between the sensor measuring $N_{sf}$ and the PID controller is labelled 0 if the communication works, 1 otherwise;
(d) The PID controller failure magnitude range is discretized into 8 equally spaced magnitude intervals, labelled from 1 to 8, representative of failure states corresponding to discrete intervals of output value belonging to \([-18, 18]\)% of the \(Q_e\) value that should be provided at \(P_o\); if the PID controller magnitude is labelled as 0, the component does not fail in \(T_{\text{miss}}\).

The values of time and magnitude and order of failure occurrence for each component are included into a sequence vector that represents a scenario. For example, the sequence vector of Figure 5 represents a scenario where the steam valve fails stuck at its maximum allowable value at a time in \([3001, 4000]\) (s) and it is the third event occurring in the sequence; the safety relief valve fails first in \([0, 1000]\) (s), with a magnitude belonging to \([0.5, 12.6]\) (kg/s); the communication between the sensor measuring \(N_t\) and the PID controller is the second failure event in the sequence and occurs in \([2001, 3000]\) (s); finally, the PID controller fails stuck in \([3001, 4000]\) (s), with a magnitude belonging to \([6, 10]\)% of the \(Q_e\) value that should be provided at \(P_o\).

The number of possible sequence vectors that arise from the MVL discretization is 100509, each one evolving towards either safe or faulty conditions. To investigate this, a Monte Carlo-driven fault injection engine is used to sample combinations of discrete times and discrete magnitudes of components failures.

The (dynamic) analysis has been performed with respect to the two constant power scenarios, 5% \(P_n\) (low power level) and 80% \(P_n\) (high power level). The system configurations considered are listed in Table 2.

The dynamic analysis shows that the same combination of components failures does not unequivocally lead to only one system end state but rather it depends on when the failures occur and with what magnitude. This is shown in Figure 6, where the frequencies of occurrence of the three system end states (“High,” “Safe,” and “Low”) are plotted for the 16 dynamic system configurations of Table 2.

---

**Figure 5: Sequence vector representing a scenario.**

**Table 2: System configurations.**

<table>
<thead>
<tr>
<th>System configurations</th>
<th>Failure of the outlet steam valve</th>
<th>Failure of the safety relief valve</th>
<th>Level sensor-PID controller communication interruption</th>
<th>Failure of the PID controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>7</td>
<td>X</td>
<td>—</td>
<td>X</td>
<td>—</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>—</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>9</td>
<td>—</td>
<td>X</td>
<td>X</td>
<td>—</td>
</tr>
<tr>
<td>10</td>
<td>—</td>
<td>X</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>—</td>
<td>—</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>—</td>
</tr>
<tr>
<td>13</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>14</td>
<td>X</td>
<td>—</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>15</td>
<td>—</td>
<td>X</td>
<td>—</td>
<td>X</td>
</tr>
<tr>
<td>16</td>
<td>X</td>
<td>X</td>
<td>—</td>
<td>X</td>
</tr>
</tbody>
</table>
Figure 6: Histograms for high power level (a) and low power level (b) of the frequencies of occurrence of the end states for each of the 16 system configurations of Table 2, simulated by sampling discrete failure times and magnitudes of components failures.

Figure 7 shows that, at high power operation, the timing of the events is quite important, because with the same system configuration but different times of failure occurrences, the system end state change. Specifically, in Figure 7(a), the safety valve fails stuck at 100% of $Q_{sf}$ after 1020 seconds and the communication between the sensor measuring $N_{rl}$ and the PID controller fails at

(i) 1052 seconds (solid line),
(ii) 1063 seconds (dashed-dotted line).

The two scenarios lead to low and high failure modes, respectively, whereas they would be considered as minimal cuts sets (MCS) in a static reliability analysis presented in Appendix A.

Figure 7(b) shows the effects of different failures magnitudes on the system end state: the safety relief valve fails stuck in its maximum position at 2000 seconds, the communication fails at 2010 seconds, and the PID controller fails at 2020 seconds with two different magnitudes:

(i) magnitude equal to 13% of the nominal $Q_{s}$ value that should be provided at 80% $P_n$ (dashed-dotted line),
(ii) magnitude equal to 12% of the nominal $Q_{s}$ value that should be provided at 80% $P_n$ (solid line).

The low power scenarios also present dynamic effects, as shown in Figure 8. In particular, Figure 8(a) shows the effects of the timing on the system end state: the safety relief valve fails stuck at $Q_{sf} = 50.5$ (kg/s) at 1005 (s) and the steam output valve fails stuck at 150% of the nominal $Q_v$ value that should be provided at 5% $P_n$ at

(i) 1046 seconds (dashed-dotted line),
(ii) 1047 seconds (solid line).

Figure 8(b) shows the effects of the order of components failure occurrence on the system end state: the safety relief valve fails stuck at $Q_{sf} = 50.5$ (kg/s) and the PID controller fails stuck at its minimum allowable value.

(i) The PID controller failure is the first failure event along the sequence of events (dashed-dotted line).
(ii) The safety relief valve failure is the first failure event along the sequence of events (solid line).

Hereafter, without loss of generality, among the system configurations of Table 2, we focus only on the classification of the PIs and Near Misses of the high level failure mode at high power level ($P_o = 80\% P_n$).

3. Near Misses Identification

The Near Misses identification is here treated as a classification problem, in which Near Misses are sorted out from the safe scenarios, among the whole set of accidental transients simulated. In practice, the PIs are first identified among the whole set of 100509 possible scenarios and, then, the Near Misses are separated out among the remaining safe scenarios.

3.1. Prime Implicants Identification. A PI is a set of variables that represents a minimal combination of accident component failures necessary for system failure and cannot be covered by a more reduced implicant [17, 18]. Note that in our case the "PIs" identification task may consider noncoherent structure functions, for which both failed and working states of the same components can lead the system to failure. In such circumstances, traditional methods, for example, based on minimal cut sets analysis, cannot be applied, whereas
dynamic reliability methods need to be applied for the identification of the PIs [33, 34].

The PIs identification among the whole set of 100509 possible scenarios is performed by means of the visual interactive method presented in [34]. The basic idea it relies on is that PIs are those scenarios with as few as possible events that are capable of leading the system into a failure state [35]; then, we first select as most important feature for the PIs identification the literal cost of the sequence vector (i.e., the number of components whose behavior is specified in the accident sequence) and then the accident sequences associated with the lowest literal cost are selected and stored as PIs. In fact,
these are the most reduced sequences (i.e., with least number of events) that cannot be covered by any other implicant, and, thus, these are PIs by definition. The selected PIs and the implicants covered by them are deleted from the set of implicants and the procedure is repeated for the remaining implicants until all are covered. By so doing, 1255 PIs are identified for the high level failure mode, covering 36128 minterms. The total computational time approximately required for the identification of the PIs is 780(s) on an Intel Core 2 Duo T9300 CPU @2.50 GHz.

3.2. The “Forward” Classification. Once the (1255) PIs for the SG high level failure mode have been identified, they are removed from the set of all possible scenarios, which is left with 64381 safe scenarios. For the identification of Near Misses among these, we resort to their definition as sequences of failure events that indeed keep the system in a safe condition but endangered (i.e., a quasifault system state). To this aim, we introduce a risk-based characterization of these remaining scenarios, calculating their associated risk, at each time instant $t$, as [23]

$$\text{Risk}(t) = p(t) \times c(t),$$

where $p(t)$ is the probability that at time $t$ the scenario can lead the system into an accidental scenario and $c(t)$ is the consequence that the developing scenario would cause to the system.

In this view, we build a functional relationship such that $p$ increases as $N_{rt}$ moves further away from the reference level $N_{ref}$ in a way that $p = 0$ if $N_{rt}$ is equal to $N_{ref}$ and $p = 1$ if $N_{rt}$ reaches $N_{high}$. Such relationship is given in (8) below, assuming that scenarios whose $N_{rt}(t)$ approaches $N_{high}$ are more prone to failure than those with $N_{rt}(t)$ close to $N_{ref}$; that is, (8) “filters out” (i.e., neglects) scenarios whose $N_{rt}(t)$ is close to $N_{ref}$ and “mines” (i.e., weighs more) scenarios whose $N_{rt}(t)$ is close to $N_{high}$:

$$p(t) = \varphi \left( \frac{N_{rt}(t) - (\mu + 5\sigma)}{\sigma} \right),$$

where $\varphi$ is the cumulative probability function of the Gaussian distribution with mean $\mu = N_{ref}$ and standard deviation $\sigma = (N_{high} - N_{ref})/2$. Figure 9 shows the trend of $p(t)$.

The consequence $c(t)$ of a scenario increases as $N_{rt}$ approaches the failure threshold $N_{high}$, and $c(t)$ can be calculated at time $t$ as [23]

$$c(t) = A(N_{RL}(t) - (\mu + 3\sigma))/(N_{RL}(t) - \mu),$$

where $A$ is the intensity coefficient that accounts for the closeness of $N_{rt}$ to the thresholds $N_{rl}, N_{sh},$ and $N_{high}$, and for the exceedance time between the first event of the failure sequence (hereafter called initiating event (IE)) and the time of exceeding the threshold: the shorter this time, the more critical the scenario. Thus, the larger $A$ is, the faster and closer $N_{rt}$ approaches a threshold; we assume $A = 100$ (no consequences) if no threshold is exceeded; $A = 200$ (low consequences) if $N_{rt}$ exceeds $N_{rl}$ after at least 2001(s) from IE or if $N_{rt}$ exceeds $N_{sh}$ after at least 3001(s) from IE; $A = 300$ (medium consequences) if $N_{rt}$ exceeds $N_{sh}$ within 2000(s) from IE, if $N_{rt}$ exceeds $N_{sh}$ and the elapsed time is in [1001, 3000] (s), and if $N_{rt}$ exceeds $N_{high}$ after at least 2001(s) from IE; $A = 400$ (catastrophic consequences) if $N_{rt}$ exceeds $N_{high}$ within 1000(s) from IE or if $N_{rt}$ exceeds $N_{high}$ and the elapsed time from IE is in [1, 2000] (s). A matrix representation of the intensity coefficient is shown in Figure 10.

By so doing, the available 64381 remaining safe scenarios are fully described at each time instant $t = 1, 2, \ldots, 4000$ [s] by their values of probability $p(t)$, consequence $c(t)$, and overall risk $r(t)$. An example of the $p(t)$, $c(t)$, and $r(t)$ evolutions for two generic trends of $N_{rt}(t)$ is shown in Figure 11. More specifically, the $N_{rt}(t)$ behaviors represented in Figure 11(a) are due to

(i) solid line: the PID controller fails at 100(s) with magnitude 4 and the safety relief valve fails at 190(s) with magnitude 2;

(ii) dashed-dotted line: the safety relief valve fails at 100(s) with magnitude 1, the communication between the sensor measuring $N_{rt}$ and the PID controller is interrupted at 136(s), and the PID controller fails at 3917(s) with magnitude 5.
Figure 10: Matrix representation of the intensity coefficient $A$.

![Figure 10](image_url)

Figure 11: Probability $p(t)$, consequences $c(t)$, and risk $r(t)$ for two sequences of events.

![Figure 11](image_url)
It is worth analysing the behavior of \( p(t), c(t), \) and, thus, \( r(t) \) (Figures 11(b), 11(c), and 11(d), resp.): all three above-mentioned functions increase as \( N_{rl}(t) \) moves further away from \( N_{rl} \) and decrease as \( N_{rl}(t) \) approaches \( N_{ref} \). The steps shown in the consequences and risk plots (around 800 [s] for the solid line scenario and around 3500 [s] for the dashed-dotted line scenario) are due to the change of the discrete consequence intensity coefficient \( A \) along the scenarios. The solid line scenario is faster than the dashed-dotted line scenario (upper plot) and, thus, the value of the parameters \( A \) for the former scenario is 400 (catastrophic consequences, see Figure 10), due to the fact that \( N_{rl}(t) \) exceeds \( N_{ref} \) within 1000 [s], whereas, \( A = 300 \) (medium consequences, see Figure 10) for the dashed-dotted scenarios, because \( N_{rl}(t) \) exceeds \( N_{ref} \) within [1001, 3000] [s]. Thus, the solid line scenario is more abrupt in its development towards failure and expected to have more catastrophic consequences, and, thus, more overall risk, than the dashed-dotted scenario, because the time between IE and the exceedance of \( N_{rl} \) is shorter (i.e, less grace time).

3.2.1. Features Selection. The identification of the Near Misses is treated as an unsupervised classification problem and addressed by clustering, where (i) the number of clusters is unknown and (ii) the features that enable the best clustering according to the risk-based characteristic profiles of \( p(t), c(t), \) and \( r(t) \) of the accidental scenarios are unknown. Unsupervised clustering, thus, entails identifying the number \( K \) of clusters in which similar scenarios can be grouped according to similar values of some scenario features. To do this, from the profiles \( p(t), c(t), \) and \( r(t) \), we extract some statistical indicators as features [36]:

\[
\begin{align*}
(1) \text{ mean value:} & \quad \mu_n = \frac{1}{T_{miss}} \sum_{t=1}^{T_{miss}} n(t), \\
(2) \text{ peak value:} & \quad \max = \max_{t=1,2,\ldots,T_{miss}} n(t), \\
(3) \text{ standard deviation:} & \quad \sigma_n = \sqrt{\frac{1}{T_{miss} - 1} \sum_{t=1}^{T_{miss}} (n(t) - \mu)^2}, \\
(4) \text{ root mean square:} & \quad \text{RMS} = \sqrt{\frac{1}{T_{miss} - 1} \sum_{t=1}^{T_{miss}} n(t)^2}, \\
(5) \text{ skewness:} & \quad SK = \frac{\sum_{t=1}^{T_{miss}} (n(t) - \mu)^3}{T_{miss} - 1},
\end{align*}
\]

(14)

where \( n(t) \) is alternatively equal to \( p(t), c(t), \) and \( r(t) \) and, thus, the total number of features is equal to \( 6 \times 3 = 18 \). Among these 6 available features, we search for those that are optimal for clustering the 64381 scenarios in Near Misses and safe scenarios.

We resort to a wrapper framework [37, 38], whereby a modified binary differential evolution (MBDE) search engine [33, 39] searches candidate groups of features sets that are fed to a \( K \)-means clustering algorithm [40]; eventually, the wrapper evolves so that among these candidate groups, the group retained is that which makes the \( K \)-means clustering algorithm perform best (most compact and separate clusters).

The idea behind the wrapper approach is shown in Figure 12. During the features search by MBDE, the \( K \)-means clustering is run on the \( N = 0.80 \times 64381 = 51505 \) (training) safe scenarios with sets of features \( F \) that are randomly selected by the MBDE algorithm. The optimal number \( (K) \) of clusters is also unknown and it is determined by looking at the clustering performance obtained by the \( K \)-means with reference to the Calinski-Harabasz (CH) index [41], which accounts for the ratio of the overall between-cluster variance (separation) and the overall within-cluster variance (compactness). The search proceeds iteratively until the CH index is maximised and the number of clusters \( K \) is fixed. Then, the results of the wrapper algorithm are evaluated on an independent test set \( (T_{test}) \), that is, the 0.2 \times 643281 = 12876 safe scenarios that have been left out during the training phase.
The CH index for a number $K$ of clusters, $k = 1, 2, \ldots, K$ is equal to [41]

$$\text{CH} = \frac{SS_b}{SS_w} \times \frac{(N \times F) - K}{K - 1},$$

where $SS_b$ is the overall between-cluster variance:

$$SS_b = \sum_{k=1}^{K} n_k \| m_k - m \|^2,$$

and $SS_w$ is the overall within-cluster variance:

$$SS_w = \sum_{k=1}^{K} \sum_{x \in k} \| x_c - m_k \|^2,$$

where $n_k$ is the number of scenarios $x_c$ assigned to the $k$th cluster, $m_k$ is the centroid of the $k$th cluster, that is, the mean of the selected features belonging to the $k$th cluster, $m$ is the mean of the selected features, and $\| m_k - m \|^2$ and $\| x_c - m_k \|^2$ are the $L^2$ norms, that is, Euclidean distances, between the two vectors.

The optimal features selection provides as best features the standard deviation of $c(t)$, the standard deviation of $r(t)$, and the root mean square of $r(t)$; the best performance is obtained with $\text{CH} = 9.35 \times 10^4$ and $K = 5$.

3.2.2. The Clustering Results. The $K = 5$ obtained clusters of the safe scenarios are shown in Figure 13 with reference to the features of mean risk ($\mu_{\text{risk}}$) and time elapsed from the instant $t_{\text{risk}}$ at which $r(t)$ starts to deviate from zero, that is, the time interval during which the system is exposed to risk. The rationale behind this choice is that the larger $\mu_{\text{risk}}$ and the longer $t_{\text{risk}}$, the more dangerous the scenarios. In Figure 13, clusters 3, 4, and 5 (triangles, crosses, and squares, resp.) are well separated, that is, the low level risk scenarios clusters are widened by the adoption of (8) for the quantification of the risk profile $r(t)$. It is possible to distinguish the scenarios having the lowest risk level from the scenarios having low risk level, and, thus, the highest risk scenarios are well separated from the lower risk scenarios. The good performance obtained when (8) is adopted instead of other $p(t)$ profiles, for example, linear probability function ($p(t) \propto N_{rl}(t)$) that would give the same importance to any level $N_{rl}$, for the quantification of the risk profile $r(t)$, is due to the fact that (8) “filters out” (i.e., neglects) scenarios whose $N_{rl}(t)$ is close to $N_{\text{ref}}$ and “mines” (i.e., weighs more) scenarios whose $N_{rl}(t)$ is close to $N_{\text{high}}$: the 332 circles in Figure 13 (listed in Appendix B) can, thus, be considered the Near Misses scenarios, that is, scenarios that incidentally keep the system into safe state, although in endangered and insecure operational conditions.

3.3. The “Backward” Approach. Once the Near Misses for the SG high level failure mode have been identified by clustering, we can search for similarities among them in terms of their multiple value sequences, that is, order and timing of event occurrences and deterministic process variables values. This “Backward” approach can lead us to finding the minimum conditions, that is, minimum $\mu_{\text{risk}}$ and minimum $t_{\text{risk}}$, that lead the system into a quasifault state. The problem can be framed as a multiobjective optimization problem (MOP) [42] that looks for the set of scenarios $\bar{x}$ to dominate any other scenarios with respect to the fitness function $f$:

$$f(x) = [f_1(\mu_{\text{risk}}), f_2(t_{\text{risk}})],$$

where $f_1$ and $f_2$ are the objectives functions of the defined MOP, that is, minimum $\mu_{\text{risk}}$ and minimum $t_{\text{risk}}$, respectively. The solution of the MOP of (19) is the Pareto set shown in Figure 14, where 12 solutions are plotted (squares lined by continuous
The coverage can be verified by, first, identifying the most similar characteristics of the sequence vectors belonging to the Near Misses cluster with the Pareto set scenarios \( \mathbf{x} \), and, then, by solving a set covering problem (SCP) [43, 44].

The most similar characteristics can be computed by coverage vectors (one for each scenario belonging to \( \mathbf{x} \)): this entails calculating the Hamming distance [45] between each sequence vectors in \( \mathbf{x} \) and each one of the other sequence vectors in the Near Misses cluster [46]. The entries of the coverage vector (in our case twelve entries, one for time, magnitude, and order of occurrence of each component failure; see Figure 5) are increased if the Hamming distance between one same entry of the considered scenario belonging to \( \mathbf{x} \) and of the Near Misses vectors is equal to zero, as shown, without loss of generality, in Figure 15 for 1 sequence vector of \( \mathbf{x} \) and only 2 Near Misses vectors.

Table 4 lists the 12 coverage vectors, where each entry is the percentage of Near Misses vectors having the same stochastic behavior of the optimal set \( \mathbf{x} \) shown in Table 3. It can be seen that, for each scenario belonging to \( \mathbf{x} \), columns 8, 11, and 12 (e.g., sensor-PID communication failure magnitude, PID failure magnitude, and PID order of failure, resp.) have the largest values of the coverage vectors: this means that the majority of the sequence vectors of the Near Misses clusters can be well represented by (only) these failures. Furthermore, the analysis of the MVL values of the scenarios belonging to \( \mathbf{x} \) (Table 3) where the largest coverage values of these columns are registered (i.e., 87%, 98.5%, and 85.2% for columns 8, 11, and 12, resp.) highlights that these failures are characterized by the same MVL values that can be summarized as follows:

(i) the failure of the communication between the sensor monitoring the \( N_{\text{rl}} \) and the PID controller;

(ii) the failure of the PID controller with magnitude belonging to \([-5, -1]\)% of the \( Q_e \) value that should be provided at \( P_{\text{opt}} \), that is, magnitude equal to 4 in MVL framework, and it is the first accident occurring along the sequence of events in over 85% of the Near Misses scenarios.

A SCP can, thus, be solved for verifying that these latest characteristics are the minimum set of stochastic event
Table 3: List of the Pareto-optimal $\mathbf{x}$ sequence vectors.

<table>
<thead>
<tr>
<th>$\mathbf{x}$</th>
<th>Steam valve failure time</th>
<th>Steam valve failure magnitude</th>
<th>Steam valve failure order</th>
<th>Safety valve failure time</th>
<th>Safety valve failure magnitude</th>
<th>Safety valve failure order</th>
<th>Sensor-PID failure time</th>
<th>Sensor-PID failure magnitude</th>
<th>Sensor-PID failure order</th>
<th>PID failure time</th>
<th>PID failure magnitude</th>
<th>PID failure order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>x</td>
<td>Steam valve failure time</td>
<td>Steam valve failure magnitude</td>
<td>Steam valve failure order</td>
<td>Safety valve failure time</td>
<td>Safety valve failure magnitude</td>
<td>Safety valve failure order</td>
<td>Sensor-PID failure time</td>
<td>Sensor-PID failure magnitude</td>
<td>Sensor-PID failure order</td>
<td>PID failure time</td>
<td>PID failure magnitude</td>
<td>PID failure order</td>
</tr>
<tr>
<td>----</td>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>--------------------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>-----------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1</td>
<td>11.4</td>
<td>11.4</td>
<td>11.4</td>
<td>8.1</td>
<td>30.4</td>
<td>0.6</td>
<td>253</td>
<td>87</td>
<td>46.4</td>
<td>1.2</td>
<td>1.5</td>
<td>0.6</td>
</tr>
<tr>
<td>2</td>
<td>30.1</td>
<td>44.3</td>
<td>271</td>
<td>1.2</td>
<td>19.9</td>
<td>38</td>
<td>19.6</td>
<td>87</td>
<td>46.4</td>
<td>68.1</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>3</td>
<td>30.1</td>
<td>44.3</td>
<td>271</td>
<td>8.1</td>
<td>19.9</td>
<td>38</td>
<td>22.3</td>
<td>87</td>
<td>14.2</td>
<td>274</td>
<td>98.5</td>
<td>14.2</td>
</tr>
<tr>
<td>4</td>
<td>30.1</td>
<td>44.3</td>
<td>22.2</td>
<td>8.1</td>
<td>19.9</td>
<td>19.9</td>
<td>253</td>
<td>87</td>
<td>12.7</td>
<td>68.1</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>5</td>
<td>30.1</td>
<td>44.3</td>
<td>271</td>
<td>28</td>
<td>30.4</td>
<td>38</td>
<td>22.3</td>
<td>87</td>
<td>46.4</td>
<td>68.1</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>6</td>
<td>30.1</td>
<td>36.1</td>
<td>271</td>
<td>1.2</td>
<td>19.9</td>
<td>19.9</td>
<td>19.6</td>
<td>87</td>
<td>13.9</td>
<td>68.1</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>7</td>
<td>30.1</td>
<td>36.1</td>
<td>271</td>
<td>8.1</td>
<td>30.4</td>
<td>0.6</td>
<td>22.3</td>
<td>87</td>
<td>46.4</td>
<td>3.3</td>
<td>1.5</td>
<td>0.60</td>
</tr>
<tr>
<td>8</td>
<td>52.4</td>
<td>44.3</td>
<td>42.2</td>
<td>28</td>
<td>19.9</td>
<td>19.9</td>
<td>253</td>
<td>87</td>
<td>12.7</td>
<td>274</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>9</td>
<td>52.4</td>
<td>36.1</td>
<td>42.2</td>
<td>1.2</td>
<td>30.4</td>
<td>19.9</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>68.1</td>
<td>1.5</td>
<td>85.2</td>
</tr>
<tr>
<td>10</td>
<td>52.4</td>
<td>36.1</td>
<td>271</td>
<td>8.1</td>
<td>9.04</td>
<td>38</td>
<td>22.3</td>
<td>87</td>
<td>14.2</td>
<td>274</td>
<td>98.5</td>
<td>14.2</td>
</tr>
<tr>
<td>11</td>
<td>52.4</td>
<td>36.1</td>
<td>42.2</td>
<td>51.2</td>
<td>19.9</td>
<td>19.9</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>1.2</td>
<td>98.5</td>
<td>85.2</td>
</tr>
<tr>
<td>12</td>
<td>52.4</td>
<td>36.1</td>
<td>271</td>
<td>51.2</td>
<td>19.9</td>
<td>38</td>
<td>22.3</td>
<td>87</td>
<td>14.2</td>
<td>1.2</td>
<td>98.5</td>
<td>14.2</td>
</tr>
</tbody>
</table>
occurrences and deterministic process variables values of $\mathbf{x}$ that exhaustively describe the scenarios belonging to the Near Misses cluster: if a Near Miss sequence vector is characterized by (at least) one of the common characteristics, this is covered by the optimal set $\mathbf{x}$. In the present application we have verified that all the scenarios belonging to the identified Near Misses cluster are covered by the minimal conditions that lead the system into a quasifault state, that is, the optimal set $\mathbf{x}$. In conclusion, the occurrence of one of the common characteristics listed above is sufficient to lead the system into endangered and insecure operational conditions.

4. Conclusions

In this paper, a risk-based clustering approach for Near Misses identification has been proposed. The approach includes a risk-based feature selection task, where by each safe scenario it is described in terms of probability, consequence, and overall risk. The optimal features set is identified by a wrapper approach based on the combination of a MBDE algorithm with $K$-means clustering. The characteristics of the Near Misses scenarios are, then, identified solving a multiobjective optimization problem and Hamming distance as a measure of similarity.

The application of the approach to a case study of IDPSA of a UTSG has shown the possibility of retrieving relevant information for risk monitoring.

Appendices

A. Static Reliability Analysis

For a static reliability analysis of the UTSG, we conservatively assume that component failures occur at the beginning of the scenario, with magnitudes equal to their extreme (either maximum or minimum) plausible values [19]. We analyze the dynamic response of the system at constant $P_o$ values ($P_o = 5\% P_n$ and $P_o = 80\% P_n$) and identify the minimal cuts sets.
Figure 17: $N_{rl}$ evolution when the PID controller output is stuck at time $t = 0$ at the minimum allowable value of $-18\%$ of nominal $Q_e$ that should be provided at $5\% P_n$ (a) and at $80\% P_n$ (b).

Figure 18: $N_{rl}$ evolution when the steam valve fails stuck at time $t = 0$ at the maximum allowable value of $150\%$ of nominal $Q_V$ that should be provided at $5\% P_n$ (a) and at $80\% P_n$ (b).

(MCS) with respect to the low and high level failure modes. Considering the binary, safe or faulty, states of the 6 components (component state is 0 if it works and 1 if it is failed), the number of possible system configurations is equal to $2^6$. However, many configurations are not detectable in constant power scenarios, for example, simultaneous occurrence of low and high power communication failures, whereas some others are not important when event occurrence timing is not considered; for example, PID and communication failures occur simultaneously, because, in this case, the feedback control output would always be the same as a stand-alone PID failure. Thus, the possible system configurations to be
considered in a static analysis with constant power is equal to 12 for each power level (Table 5).

To identify the system MCS, the different system configurations of Table 5 have been simulated by the SIMULINK model, at low and at high (constant) power levels. It turns out that the MCSs for the high level failure mode are the same at both power levels (Figure 16): the failures of the PID controller at its minimum values (i.e., −18% of the nominal $Q_e$ that should be provided at $P_o$) and of the steam valve at its maximum value (i.e., 150% of the nominal $Q_v$ value that should be provided at $P_o$) are two first-order MCS. The $N_{rl}$ evolutions when these MCSs occur are shown in Figures 17 and 18.

The analysis of the low level failure mode provides different MCSs at different $P_o$. At 5% $P_o$, there are three first-order MCSs represented by the following: (i) safety valve fails stuck at the maximum allowable value: that is, $Q_{sf} = 50.5$ (kg/s); (ii) steam valve fails stuck closed; (iii) PID
controller fails stuck at its maximum values, (i.e., 18% of the nominal $Q_e$ value that should be provided at 5% $P_n$). The $N_{rl}$ evolution when these MCSs occur and the relative FT are shown in Figures 19 and 20, respectively.

At 80% $P_n$, three MCSs are found: (i) a second-order MCS that combines the failure of the safety relief valve at its maximum allowable value: that is, $Q_{sf} = 50.5$ (kg/s), and the failure of the communication, (ii) the steam valve failure in a closed position, and (iii) the PID controller fails at its maximum value (i.e., 18% of the nominal $Q_e$ value that should be provided at 80% $P_n$). The $N_{rl}$ evolution when these MCSs occur and the relative FT are shown in Figures 21 and 22, respectively.

### B. Near Misses Sequence Vector Scenarios

See Table 6.

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CET</td>
<td>Continuous event tree</td>
</tr>
<tr>
<td>CH</td>
<td>Calinski-Harabasz index</td>
</tr>
<tr>
<td>DBA</td>
<td>Design basis accident</td>
</tr>
<tr>
<td>DET</td>
<td>Dynamic event tree</td>
</tr>
<tr>
<td>DSA</td>
<td>Deterministic safety analysis</td>
</tr>
<tr>
<td>ET</td>
<td>Event tree</td>
</tr>
<tr>
<td>FT</td>
<td>Fault tree</td>
</tr>
<tr>
<td>IDPSA</td>
<td>Integrated deterministic and probabilistic safety analysis</td>
</tr>
<tr>
<td>IE</td>
<td>Initiating event</td>
</tr>
<tr>
<td>MBDE</td>
<td>Modified binary differential evolution</td>
</tr>
<tr>
<td>MCS</td>
<td>Minimal cuts set</td>
</tr>
<tr>
<td>MOP</td>
<td>Multiobjective optimization problem</td>
</tr>
<tr>
<td>MVL</td>
<td>Multiple-valued logic</td>
</tr>
<tr>
<td>NPP</td>
<td>Nuclear power plant</td>
</tr>
<tr>
<td>PI</td>
<td>Prime implicants</td>
</tr>
<tr>
<td>PSA</td>
<td>Probabilistic safety analysis</td>
</tr>
<tr>
<td>SCP</td>
<td>Set covering problem</td>
</tr>
<tr>
<td>SG</td>
<td>Steam generator</td>
</tr>
<tr>
<td>TH</td>
<td>Thermal-hydraulics</td>
</tr>
<tr>
<td>UTSG</td>
<td>U-tube steam generator</td>
</tr>
</tbody>
</table>

### Symbols

- $p$: Probability that the developing scenario is an accidental scenario
- $c$: Consequence that the developing scenario can cause to the system
- $r$: Overall risk of the developing scenario
- $t$: Time instant
- $p(t)$: Probability that at time $t$ the scenario can lead the system into an accidental scenario
- $c(t)$: Consequence that at time $t$ the developing scenario is predicted to cause to the system
- $r(t)$: Overall risk of the developing scenario at time $t$
- $Q_{e}$: Flow rate of fresh feed-water entering the steam generator
- $P_o$: Operating power
- $P_n$: Nominal power
- $Q_{v}$: Flow rate of dry steam exiting the steam generator
- $N_{rl}$: Narrow range steam generator water level
- $W_{rl}$: Wide range steam generator water level
- $T_n$: Time constant for the $N_{rl}$ dynamics
- $Q_{ef}$: Flow rate of incoming water in steam generator tube bundle region
- $T_h$: Time constant for the water mass transportation dynamics
- $r$: Time constant for the feed-water valve dynamics
- $Q_{GV}$: Flow rate of steam-water mixture exiting the steam generator tube bundle region
- $T_{g}$: Time constant for the dynamics relating $Q_v$ to $Q_{GV}$
- $F_{g}$: Constant in the nonminimum phase term of the dynamics relating $Q_v$ to $Q_{GV}$
- $T_{ini}$: Time constant for the $W_{rl}$ dynamics
<table>
<thead>
<tr>
<th>Near Miss</th>
<th>$T_s$</th>
<th>$M_s$</th>
<th>$O_s$</th>
<th>$T_{sa}$</th>
<th>$M_{sa}$</th>
<th>$O_{sa}$</th>
<th>$T_c$</th>
<th>$M_c$</th>
<th>$O_c$</th>
<th>$T_p$</th>
<th>$M_p$</th>
<th>$O_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>41</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>42</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>43</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>44</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>45</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>46</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Near Miss</td>
<td>$T_s$</td>
<td>$M_1$</td>
<td>$O_1$</td>
<td>$T_{sa}$</td>
<td>$M_{sa}$</td>
<td>$O_{sa}$</td>
<td>$T_r$</td>
<td>$M_r$</td>
<td>$O_r$</td>
<td>$T_p$</td>
<td>$M_p$</td>
<td>$O_p$</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>47</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>49</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>50</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>51</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>53</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>54</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>55</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>57</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>58</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>59</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>60</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>61</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>62</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>63</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>65</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>66</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>67</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>68</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>69</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>70</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>72</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>73</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>74</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>75</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>76</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>77</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>78</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>79</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>81</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>82</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>83</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>85</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>86</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>87</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>88</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>89</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>90</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>91</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>92</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>93</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>94</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 6: Continued.

<table>
<thead>
<tr>
<th>Near Miss</th>
<th>$T_s$</th>
<th>$M_s$</th>
<th>$O_s$</th>
<th>$T_{sa}$</th>
<th>$M_{sa}$</th>
<th>$O_{sa}$</th>
<th>$T_c$</th>
<th>$M_c$</th>
<th>$O_c$</th>
<th>$T_p$</th>
<th>$M_p$</th>
<th>$O_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>96</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>97</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>98</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>99</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>102</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>103</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>104</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>105</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>106</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>107</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>108</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>109</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>110</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>111</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>112</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>113</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>114</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>115</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>116</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>117</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>118</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>119</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>120</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>121</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>122</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>123</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>124</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>125</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>126</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>127</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>129</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>130</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>131</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>132</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>133</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>134</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>135</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>136</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>137</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>138</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>139</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>140</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Near Miss</td>
<td>( T_s )</td>
<td>( M_s )</td>
<td>( O_s )</td>
<td>( T_{sa} )</td>
<td>( M_{sa} )</td>
<td>( O_{sa} )</td>
<td>( T_p )</td>
<td>( M_p )</td>
<td>( O_p )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>177</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>179</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>182</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>185</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6: Continued.

<table>
<thead>
<tr>
<th>Near Miss</th>
<th>$T_s$</th>
<th>$M_s$</th>
<th>$O_s$</th>
<th>$T_{sa}$</th>
<th>$M_{sa}$</th>
<th>$O_{sa}$</th>
<th>$T_c$</th>
<th>$M_c$</th>
<th>$O_c$</th>
<th>$T_p$</th>
<th>$M_p$</th>
<th>$O_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>189</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>190</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>191</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>192</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>193</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>194</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>195</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>196</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>197</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>198</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>199</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>200</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>201</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>202</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>203</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>204</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>205</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>206</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>207</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>208</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>209</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>210</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>211</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>212</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>213</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>214</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>215</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>216</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>217</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>218</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>219</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>220</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>221</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>222</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>223</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>224</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>225</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>226</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>227</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>228</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>229</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>231</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>232</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>233</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>234</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>235</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>236</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>237</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 6: Continued.

<table>
<thead>
<tr>
<th>Near Miss</th>
<th>$T_s$</th>
<th>$M_s$</th>
<th>$O_s$</th>
<th>$T_{sa}$</th>
<th>$M_{sa}$</th>
<th>$O_{sa}$</th>
<th>$T_p$</th>
<th>$M_p$</th>
<th>$O_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>238</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>239</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>240</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>241</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>242</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>243</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>244</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>245</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>246</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>247</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>248</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>249</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>250</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>251</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>252</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>253</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>254</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>255</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>256</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>257</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>258</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>259</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>260</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>261</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>262</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>263</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>264</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>265</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>266</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>267</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>268</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>269</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>270</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>271</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>272</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>273</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>274</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>275</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>276</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>277</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>278</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>279</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>280</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>281</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>282</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>283</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>284</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>285</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Table 6: Continued.

<table>
<thead>
<tr>
<th>Near Miss</th>
<th>$T_s$</th>
<th>$M_s$</th>
<th>$O_s$</th>
<th>$T_{sa}$</th>
<th>$M_{sa}$</th>
<th>$O_{sa}$</th>
<th>$T_p$</th>
<th>$M_p$</th>
<th>$O_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>286</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>287</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>288</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>289</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>290</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>291</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>292</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>293</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>294</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>295</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>296</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>297</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>298</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>299</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>300</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>301</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>302</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>303</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>304</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>305</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>306</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>307</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>308</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>309</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>310</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>311</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>312</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>313</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>314</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>315</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>316</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>317</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>318</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>319</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>320</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>321</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>322</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>323</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>324</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>325</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>326</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>327</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>328</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>329</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>330</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>331</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>332</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 21: (a) $N_{sl}$ evolution when simultaneously the safety relief valve fails stuck with $Q_{sf} = 50.5$ (kg/s) and the communication fails; (b) the steam valve fails stuck closed; (c) the PID controller fails stuck at 18% of the nominal $Q_e$ that should be provided at 80% $P_{in}$.

$x$: System state
$x'$: Derivative of system state
$N_{ref}$: Narrow range steam generator water level at a reference position
$N_{high}$: Automatic reactor trip threshold
$N_{low}$: Turbine trip threshold
$N_{sl}$: First prealarm automatic reactor trip threshold
$N_{sl}$: First prealarm turbine trip threshold
$N_{sh}$: Second prealarm automatic reactor trip threshold
$N_{st}$: First prealarm turbine trip threshold
$Q_{pid}$: Water flow rate provided by PID controller
$Q_{sf}$: Water flow rate removed by safety valve
$T_{miss}$: Mission time
$t_{mvl}$: Time steps in MVL discretization
$\phi$: Cumulative probability function of the Gaussian distribution.
The authors declare no conflicts of interest.

**Conflict of Interests**

The authors declare no conflict of interests.

**References**


