Measuring Risk-Importance in a Simulation-Based PRA Framework - Part I: Mathematical Framework

D. Mandelli, Z. Ma, C. Parisi, D. Maljovec, A. Alfonsi, C. Smith

Idaho National Laboratory (INL), 2525 Fremont Ave, 83402 Idaho Falls (ID), USA

Abstract

Risk importance measures are indexes that are used to rank systems, structures and components (SSCs) using risk-informed methods. The most used/known measures are: Risk Reduction Worth (RRW), Risk Achievement Worth (RAW), Birnbaum (B) and Fussell-Vesely (FV). Once obtained from classical Probabilistic Risk Analysis (PRA) analyses, these risk measures can be effectively employed to optimize component testing and maintenance. In contrast to classical PRA methods, Dynamic PRA methods couple stochastic methods with safety analysis codes to determine risk associate to complex systems such as nuclear plants. Compared to classical PRA methods, they can evaluate with higher resolution the safety impact of timing and sequencing of events on the accident progression. The objective of this paper is to present a series of algorithms that can be used to determine classical risk importance measures (RRW, RAW, B and FV) along with newly developed ones from a Dynamic PRA analysis.

Keywords: Importance Measures, Dynamic PRA, Probabilistic Risk Assessment

1. Introduction

Risk Importance Measures (RIMs) [1] are indexes that are used to rank systems, structures and components (SSCs) based on their contribution to the overall risk. The most used measures [2] are: Risk Reduction Worth (RRW), Risk Achievement Worth (RAW), Birnbaum (B) and Fussell-Vesely (FV).

Typically, this ranking is performed in a classical PRA framework, where risk is determined by considering probability associated to the minimal cut-sets generated by static logic structures [3] (e.g., Event-Trees, Fault-Trees). In a classical PRA analysis, each SSC is represented by a set of basic events; as an example emergency diesel generators can be represented by two basic events: failure to start and failure to run.

The risk measures associated to each basic event are calculated from the generated cut-sets by determining:

- The nominal risk

- The increased risk assuming basic event failed
- The reduced risk assuming basic event perfectly reliable

In this context, the Nuclear Regulatory Commission (NRC) has issued the 10CFR50.69 document [] designed to guide plant owners to perform a risk-informed categorization and treatment of SSCs in order to reduce operating and maintenance costs while preserving acceptable risk levels. The described categorization is based on a set of risk importance measures obtained from the plant classic PRA models.

In contrast to classical PRA methods, Dynamic PRA methods [4] couple stochastic methods (e.g., RAVEN [5], ADAPT [6], ADS [7], MCDET [8]) with safety analysis codes (RELAP5-3D [9], MELCOR [10], MAAP [11]) to determine risk associate to complex systems such as nuclear plants. Accident progression is thus determined by the simulation code and not set a-priori by the user. The advantage of this approach, compared to classical PRA methods, is that a higher fidelity of the results can be achieved since:

- No assumption of timing/sequencing of events is set by the user but dictated by the accident evolution
- No success criteria are defined but instead, the simulation stops if either a fail or a success state are reached
- There is no need to compute convolution integrals in order to specify probability of basic events that temporally depends from other basic events.

The scope of this paper is to present a method to determine classical RIMs from Dynamic PRA data. Few test cases will be presented in order to show how the calculation is performed. In addition, new margin-centric RIMs that better capture the continuous aspect of a Dynamic PRA approach will be presented.

2. Classical RIMs

The SAPHIRE PRA code can calculate the following seven different basic event importance measures for each basic event for the respective fault tree, accident sequence, or end state:

- Fussell-Vesely (FV)
- Risk Increase Ratio (RIR)
- Risk Increase Difference (RID)
- Risk Reduction Ratio (RRR)
- Risk Reduction Difference (RRD)
- Birnbaum (B)

- Uncertainty Importance

The most used importance measures are Fussell-Vesely, Risk Increase Ratio, Risk Reduction Ratio, and Birnbaum. The Fussell-Vesely importance measure indicates the fraction of the minimal cut set upper bound (or sequence frequency, core damage frequency) contributed by the cut sets containing the interested basic event. It is calculated in SAPHIRE Version 8 with the following equation: FV = F(i) / F(x) where:

- F(x) = value of all the minimal cut sets evaluated with the basic event probabilities at their mean value
- F(i) = value of all the minimal cut sets that contain the interested basic event i.

The Risk Increase Ratio or Risk Increase Difference importance measure indicates the increase (in relative ratio changes or in actual differences) of the minimal cut set upper bound (or sequence frequency, core damage frequency) if the interested basic event always occurred (i.e., the basic event failure probability is 1.0). The Risk Increase Ration importance is often called Risk Reduction Worth (RRW) in industry. The risk reduction importance measures are calculated in SAPHIRE Version 8 with the following equation: RIR (or RAW)=F(1)/F(x) RID=F(1)-F(x) where F(x) = value of all the minimal cut sets evaluated with the basic event probabilities at their mean value. F(1) = value of all the minimal cut sets evaluated with the interested basic event probability set to 1.0. The Risk Reduction Ratio or Risk Reduction Difference importance measure indicates the reduction (in relative ratio changes or in actual differences) of the minimal cut set upper bound (or sequence frequency, core damage frequency) if the interested basic event never occurred (i.e., the basic event failure probability is 0.0). The Risk Reduction Ratio importance is often called Risk Achievement Worth (RAW) in industry. The risk increase importance measures are calculated in SAPHIRE Version 8 with the following equation: RRR (or RRW)=F(x)/F(0) RRD=F(x)-F(0) where F(x) = value of all the minimal cut sets evaluated with the basic event probabilities at their mean values. F(0) = value of all the minimal cut sets evaluated with the interested basic event probability set to 0.0. The Birnbaum importance measure is an indication of the sensitivity of the minimal cut set upper bound (or sequence frequency, core damage frequency) with respect to the interested basic event. It is calculated by the following equation: B=F(1)-F(0) where F(1) = value of all the minimal cut sets evaluated with the interested basic event probability set to 1.0. F(0) = value of all the minimal cut sets evaluated with the interested basic event probability set to 0.0. The Uncertainty Importance measure is an indication of the contribution of the interested basic events uncertainty to the total output uncertainty. This importance measure is not widely used and is not discussed in further detail. As a simple example, lets look at the importance measures of basic event LPI-XHE-XM-ERRORHL in LLOCA event tree. The value of all the minimal cut sets evaluated with the basic event probabilities

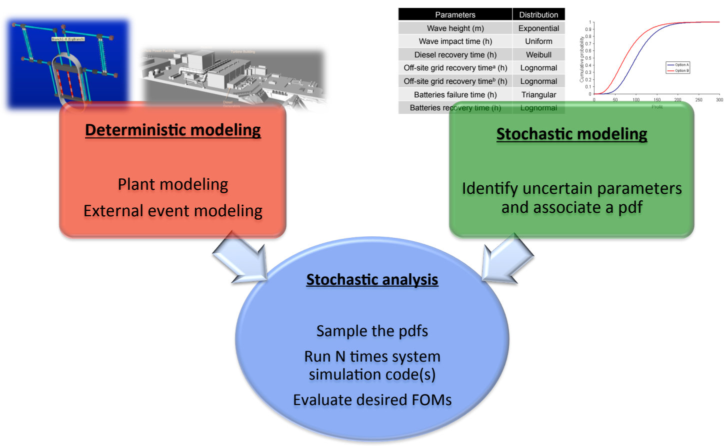


Figure 1: Overview of the RISMC approach

at their mean values for LLOCA event tree is 2.036E-8/year. There is only one cut set that contain the interested basic event, LPI-XHE-XM-ERRORHL, whose mean probability is 5.00E-3 (refer to the table below for top 20 cut sets of LLOCA event tree).

3. RISMC Approach to PRA

The RISMC approach [12] employs both deterministic and stochastic methods in a single analysis framework (see Figure 1). In the deterministic method set we include:

- Modeling of the thermal-hydraulic behavior of the plant [13, 14]
- Modeling of external events such as flooding [15]
- Modeling of the operators responses to the accident scenario [16]

Note that deterministic modeling of the plant or external events can be performed by employing specific simulator codes but also surrogate models [**?**], known as reduced order models (ROM). ROMs would be employed in order to decrease the high computational costs of employed codes. In addition, multi-fidelity codes can be employed to model the same system; the idea is to switch from low-fidelity to high-fidelity code when higher accuracy is needed (e.g., use low-fidelity codes for steady-state conditions and high-fidelity code for transient conditions)

In the stochastic modeling we include all stochastic parameters that are of interest in the PRA analysis such as uncertain parameters and stochastic failure of system/components. As mentioned earlier, the RISMC approach heavily relies on multi-physics system simulator codes (e.g., RELAP5-3D [9]) coupled with stochastic analysis tools (e.g., RAVEN [**?**]). From a PRA point of view, this type of simulation can be described by using two sets of variables:

- $\boldsymbol{c} = \boldsymbol{c}(t)$ represents the status of components and systems of the simulator (e.g., status of pumps and valves)
- $\boldsymbol{\theta} = \boldsymbol{\theta}(t)$ represents the temporal evolution of a simulated accident scenario, i.e., $\boldsymbol{\theta}(t)$ represents a single simulation run. Each element of $\boldsymbol{\theta}$ can be for example the values of temperature or pressure in a specific node of the simulator nodalization.

From a mathematical point of view, a single simulator run can be represented as a single trajectory in the phase space. The evolution of such a trajectory in the phase space can be described as follows:

$$\begin{cases} \dfrac{\partial \boldsymbol{\theta}}{\partial t} = \boldsymbol{\Xi}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{s}, t) \\ \\ \dfrac{\partial \boldsymbol{c}}{\partial t} = \boldsymbol{\Gamma}(\boldsymbol{\theta}, \boldsymbol{c}, \boldsymbol{s}, t) \end{cases} \tag{1}$$

where:

- $\boldsymbol{\Xi}$ is the actual simulator code that describes how $\boldsymbol{\theta}$ evolves in time
- $\boldsymbol{\Gamma}$ is the operator which describes how c evolves in time , i.e., the status of components and systems at each time step
- $\boldsymbol{s}$ is the set of stochastic parameters.

Starting from the system located in an initial state, $\boldsymbol{\theta}(t = 0) = \boldsymbol{\theta}(0)$, and the set of stochastic parameters (which are generally generated through a stochastic sampling process), the simulator determine at each time step the temporal evolution of $\boldsymbol{\theta}(t)$. At the same time, the system control logic determines the status of the system and components $\boldsymbol{c}(t)$.

By using the RISMC approach, the PRA analysis is performed by []:

1. Associating a probabilistic distribution function (pdf) to the set of parameters $\boldsymbol{s}$ (e.g., timing of events)
2. Performing stochastic sampling of the pdfs defined in Step 1
3. Performing a simulation run given $\boldsymbol{s}$ sampled in Step 2, i.e., solve the system of equations **??**
4. Repeating Steps 2 and 3 M times and evaluating user defined stochastic parameters such as core damage (CD) probability (P_{CD}).

Note that s includes not only uncertain parameters characteristic of the simulator (e.g., pipe friction coefficients) but also the set Basic Events (BEs) associated to the considered components.

The goal of measuring components risk importance is to identify the components that contribute the most to the system/plant overall risk.

The objective of this identification process once completed is that testing and maintenance procedures can be directed towards more risk-significant components or they can be replaced with more reliable models.

3.1. RISMC Approach and Classical PRA

In a classical PRA framework, each BE has a unique probability value associated to it while in a dynamic PRA one BEs each BE has a probability distribution function (pdf) associated to it. This pdf can be discrete in nature (e.g.,a bernoulli distribution) or continuous (e.g., exponential).

As an example lets consider two basic events associated to the emergency diesel generators (EDGs) of a nuclear power plant: EDG failure to start (EDG_FS) and EDG failure to run (EDG_FS). In a classical PRA framework two probability values would be associated to each basic event: p_{EDG_FS} and p_{EDG_FR}. In a dynamic PRA framework two pdfs would be associated to each basic event:

- $EDG\_FS \sim \text{Bern}(p_{EDG\_FS})$ (Bernoulli distribution)
- $EDG\_FR \sim \text{Exp}(\lambda_{EDG\_FR})$ (Exponential distribution).

When comparing Dynamic vs. Classical PRA approaches note the following:

- EDG_FS has the identical statistical model in the two approaches
- EDG_FR has different statistical models; however, if we set

$$p_{EDG\_FR} = \int_0^{MT} \lambda_{EDG\_FR} \cdot e^{-\lambda_{EDG\_FR} t} dt \tag{2}$$

  where MT is the EDG mission time, then the two models are identical from a statistical perspective.

An additional methodological difference among classical and dynamic PRA is the modeling of sequencing and timing of events. In classical PRA this is typically performed using Event-Trees (ETs) where sequence and timing of events are set by the analysis prior the analysis. An example of ET is shown in Fig. 2 for large Loss Of Coolant Accident (LOCA): successful outcome of each ET branch is guaranteed only if the accumulator, low-pressure injection (LPI) and low-pressure recirculation (LPR) systems successfully perform their function. Each ET branch corresponds to a possible accident scenario while each branching point corresponds to the successful or failed activation of a system (accumulator, LPI and LPR) The ET construction requires the definition of a set of acceptance criteria (e.g., collapsed level greater than 1/3 of core height) and a set of success criteria for each system involved in the accident progression. This criteria are determined by the analysis and are backed up by a set of thermo-hydraulic calculations.

In a dynamic PRA method, timing and sequencing of events are uniquely dictated by the system control logic and by the set of stochastic parameters $\boldsymbol{s}$,i.e., the construction of the ET is replaced by coding the plant control logic (e.g., system activation points and activation rules). In addition, acceptance criteria and success criteria are incorporated into the physics model of the code Back to the large LOCA scenario, in a dynamic framework it would be modeled by:

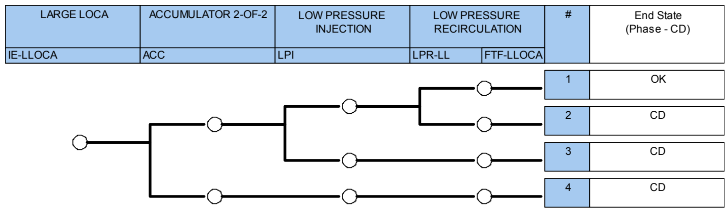


Figure 2: Large LOCA ET.

- employing a system simulator code (e.g., RELAP5-3D)
- three stochastic parameters:
  - accumulator system: failure on demand (Bernoulli distribution)
  - LPI system: failure to run (Exponential distribution)
  - LPR system: failure to run (Exponential distribution)
- the condition to end a code simulation run and its corresponding outcome:
  - OK outcome: mission time (e.g., 24 hours)
  - Fail outcome: max core temperature greater than 2200 F

Note that a large discrepancies among classical and dynamic PRA methods can occur for large sequence of events coupled with system dynamics. Assuming two events A and B occur in sequence and time of activation of each of them is a stochastic variable ($t_A \sim pdf_A(t)$ and $t_B \sim pdf_B(t)$), the actual activation time T_B of system B is a stochastic variable given by the sum of t_A and t_B ($T_B = t_A + t_B$). In a classical PRA framework, the distribution of T_B (i.e., $pdf_{A+B}(t)$) can be determined by solving the convolution integral:

$$pdf_{A+B}(t) = \int_{-\infty}^{\infty} pdf_B(t-\tau) pdf_A(\tau) d\tau \qquad (3)$$

This convolution integral get more complex when accident progression include a large number of events and system dynamics affect timing of events.

4. RAVEN

The Risk Analysis and Virtual ENviroment (RAVEN[1]) [5, 17] is a flexible and multi-purpose uncertainty quantification, regression analysis, probabilistic risk assessment, data analysis and model optimization framework. Depending on the tasks to be accomplished and on the probabilistic characterization

[1] Official website: `https://raven.inl.gov`,
GITHUB repository: `https://github.com/idaholab/raven`

of the problem, RAVEN perturbs (e.g., Monte-Carlo, latin hypercube, reliability surface search) the response of the system under consideration by altering its own parameters. The system is modeled by third party software (e.g., RELAP5-3D [9], MELCOR [10]) and accessible to RAVEN either directly (software coupling) or indirectly (via input/output files). The data generated by the sampling process is analyzed using classical statistical and more advanced data mining approaches. RAVEN also manages the parallel dispatching (i.e. both on desktop/workstation and large High Performance Computing machines) of the software representing the physical model. RAVEN heavily relies on artificial intelligence algorithms to construct surrogate models of complex physical systems in order to perform uncertainty quantification, reliability analysis (limit state surface) and parametric studies.

By statistical analysis we include:

- Sampling of codes, either stochastic, e.g., Monte-Carlo [18] and Latin Hypercube Sampling (LHS) [19], deterministic (e.g., grid and Dynamic Event Tree (DET) [20, 21]) or adaptive [22, 23]
- Generation of ROMs [24], also known as Surrogate models
- Post-processing of the sampled data and generation of statistical parameters (e.g., mean, variance, covariance matrix).

Figure 3 shows a general overview of the elements that comprise the RAVEN statistical framework:

- Model: it represents the pipeline between input and output space. It comprises both codes (e.g., RELAP5-3D [9]) and also ROMs
- Sampler: it is the driver for any specific sampling strategy, e.g., Monte-Carlo, LHS, DET [25, 26])
- Database: the data storing entity
- Post-processing module: the module that performs statistical analyses and visualizes results.

5. Classical RIMs in a Dynamic PRA Context

In a Dynamic PRA environment, R_0 is obtained (e.g., through Monte-Carlo sampling) by:

1. Running N simulation (e.g., RELAP5 runs)
2. Counting the number N_{CD} of simulations that lead to core damage (CD) condition
3. Calculating $R_0 = \frac{N_C D}{N}$

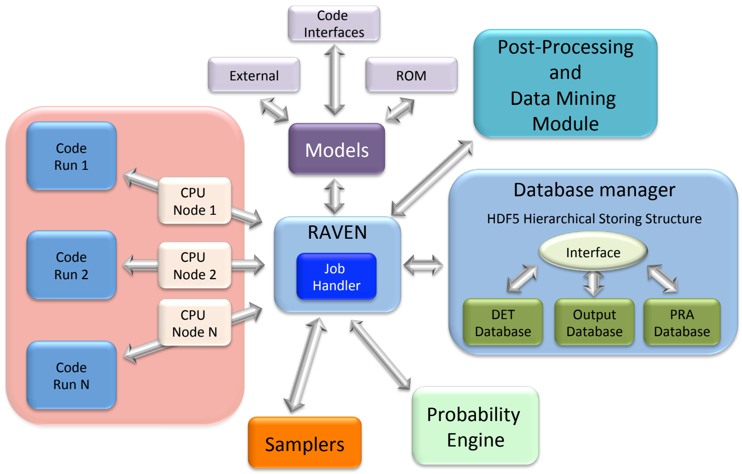


Figure 3: Overview of RAVEN statistical framework components

Note that while basic events in classical PRA are mainly Boolean, in a Dynamic PRA environment the sample parameters can be, not only Boolean, but more often continuous. As an example, let consider two basic events:

- Emergency Diesel Generator (EDG) failure to start, and,
- EDG failure to run

In classical PRA analyses, a probability value is associated to each basic event. On the other side, in a Dynamic PRA framework, a Bernoulli distribution could be associated to the first basic event and a continuous distribution (e.g., exponential distribution) could be associated to the second basic event. At this point a challenge arises: the determination of R_i^- and R_i^+ for each sampled parameter; two possible approaches can be followed :

1. Perform a Dynamic PRA for R_0 and each R_i^- and R_i^+
2. Determine an approximated value of R_i^- and R_i^+ from the simulation runs generated to calculate R_0

Regarding Approach 1, given the computational costs of each Dynamic PRA, it is unfeasible to determine R_i^- and R_i^+ for each sampled parameter. In fact, if we consider M sample parameters (i.e., S basic events), then the risk importance analysis would require $2S + 1$ Dynamic PRA analyses.

Regarding Approach 2, a method (implemented in RAVEN as an internal post-processor) was developed and it is here presented. This method requires an input from the user:

- Range, I_i^-, of the variable s_i that can be associated to “basic event with component perfectly reliable”

- Range, I_i^+, of the variable s_i that can be associated to "basic event in a failed status"

Given this kind of information, it is possible to calculate R_i^+ and R_i^- as follows:

$$R_0 = \frac{N_{CD}}{N} \tag{4}$$

$$R_i^+ = \frac{N_{CD,s_i \in I_i^+}}{N} \tag{5}$$

$$R_i^- = \frac{N_{CD,s_i \in I_i^-}}{N} \tag{6}$$

where:

- $N_{CD,s_i \in I_i^+}$ is the number of simulations leading to core damage and with parameter $s_i \in I_i^+$
- $N_{CD,s_i \in I_i^-}$ is the number of simulations leading to core damage and with parameter $s_i \in I_i^-$

Note that this approach has an issue related to the choices of I_i^+ and I_i^-. Depending on their values, R_i^+ and R_i^- might change accordingly. In addition, the statistical error associated to the estimates of R_i^+ and R_i^- also changes. An example is shown in Figure 1 for both cases (discrete and continuous) of a basic event x_i represented as a stochastic variable which is sampled (e.g., through a Monte-Carlo process) for each simulation run. Lets consider the continuous case and assume s_i correspond to the basic event "EDG failure to run". The user might impose the following in order to determine R_i^+ and R_i^-:

- $I_i^- = [T_i^-, \infty]$ where T_i^- may be set equal to the simulation mission time (e.g., 24 hours). This implies that a sampled value for EDG failure to run greater than 24 hours implies that the EDG actually does not fail to run (reliability equal to 1.0)
- $I_i^+ = [0, T_i^+]$ where T_i^- may be set to an arbitrary small value (e.g., 5 min). This implies that a sampled value for EDG failure to run smaller than 5 min implies a reliability equal to 0.0

Note that while the definition of I_i^- is perfectly reasonable, one would argue that a smaller interval should be chosen for I_i^+ (e.g., 30 seconds or less). Recall that ideally, a value of $s_i = 0.0$ should be theoretically chosen (and not an interval); however, given the nature of the distribution this is not allowed. Given the nature of the problem, we are bound to choose an interval I_i^+:

- A small interval in the neighbor of $s_i = 0.0$ would lead to a value of R_i^+ close to the theoretical one. However, the number of actual sampled values falling in I_i^+ would be very small, i.e., large stochastic error.

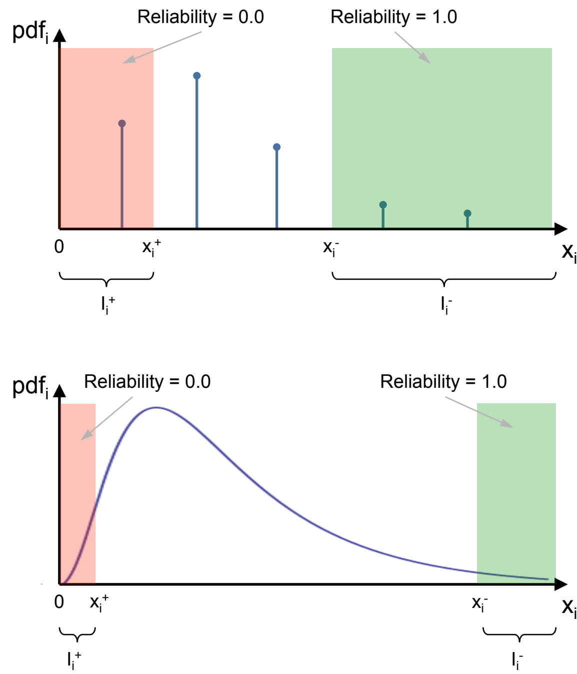


Figure 4: Treatment of discrete (top) and continuous (bottom) stochastic variables for reliability purposes.

- A large interval in the neighbor of $s_i = 0.0$ would lead to a value of R_i^+ far from the theoretical one. However, the number of actual sampled values falling in I_i^+ would be very high, i.e., small stochastic error.

A solution to the large statistical error associated to a very small interval I_i^+ can be solved by employing different sampling algorithms other than the classical Monte-Carlo one. As an example, a better resolution of the final value for R_i^+ can be achieved by sampling uniformly the range of variability of x_i and associate an importance weight to each sample. At this point the counting variable N_{CD} is weighted by the weight of each sample. By sampling uniformly the range of variability of x_i, the number of samples in the interval I_i^+ would be significantly higher.

6. Test Examples

In order to better understand the results obtained in this section it is worth to illustrate a link between classical PRA and RISMC approach. Let's consider a system that is composed by two components (i.e, A and B) in a series configuration where each component has a failure probability (i.e., p_A and p_B respectively) as shown in Fig. 5.

In a classical PRA framework such system can be modeled using a FT method that is composed by two basic events: A failed and B failed. System failure would be represented by a single "AND" gate that combine the two basic events as shown in Fig. 5.

In a RISMC approach, such system would be modeled by using two stochastic parameters (i.e., var_A and var_B) with a Bernoulli distribution Bern(p) associated to each of them: $var_A \sim \text{Bern}(p_A)$ and $var_B \sim \text{Bern}(p_B)$. The model that emulates system response would simply implement the "AND" logic of the var_A and var_B. In order to determine system failure probability a numerical integration has to be performed in a 2-dimensional (a dimension for each stochastic parameter).

Two possible sampling strategies can be followed:

- Monte-Carlo: generate N samples and count the the number of samples that lead to system failure

- Grid: partition the 2-dimensional space into a Cartesian grid; generate a sample for each partition and associate a probability weight w to each sample. This weight can be determined by integrating the pdf $pdf(A, B) = \text{Bern}(p_A)\,\text{Bern}(p_B)$ in each partition. In this specific case, since each stochastic parameter has two possible outcomes (i.e. 0 and 1), the space has been partitioned into 4 regions as shown in Fig. 6. Each cell of Fig. 6 has indicated the system outcome: system failure (F) or system success (OK).

The Monte-Carlo would require a large number of samples in order to decrease the statistical error associated to system failure probability. On the other hand a Grid sampler would determine the exact value of system failure probability with only 4 samples:

1. sample 1: $A = 0$ and $B = 0$ (bottom left cell of Fig. 6), $w_1 = (1 - p_A)(1 - p_B)$
2. sample 2: $A = 1$ and $B = 0$ (top left cell of Fig. 6), $w_2 = p_A(1 - p_B)$
3. sample 3: $A = 0$ and $B = 1$ (bottom right cell of Fig. 6), $w_3 = (1 - p_A)p_B$
4. sample 4: $A = 1$ and $B = 1$ (top right cell of Fig. 6), $w_4 = p_A p_B$

Note that each sample/cell of Fig. 6 that leads to system failure corresponds to a specific cut-set

- Cut-set 1 (CS1) corresponds to sample 2; $p_{CS1} = p_A$

- Cut-set 2 (CS2) corresponds to sample 3; $p_{CS2} = p_B$

- Cut-set 3 (CS3) corresponds to sample 4; $p_{CS3} = p_A p_B$

Hence, the two methods (classical and PRA) would provide identical results.

Observe now that the number of samples required for M stochastic parameters (assuming they are all distributed with a Bernoulli distribution) would be equal to 2^M. Thus this strategy can be employed for a small value of M.

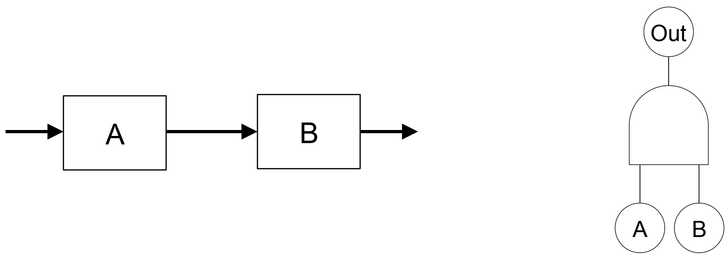


Figure 5: Components A and B in a series configuration (left) and its associated Fault-Tree (right).

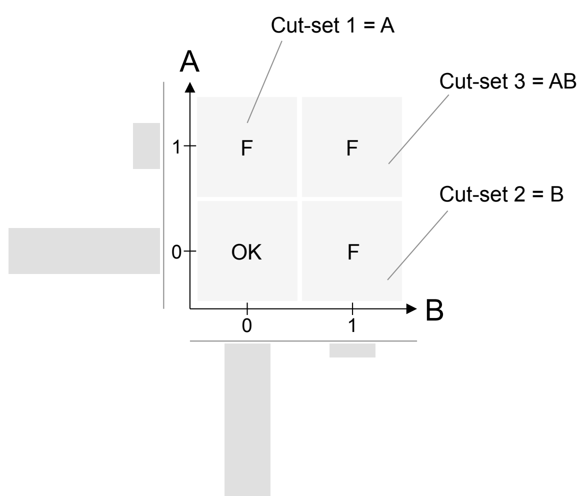


Figure 6

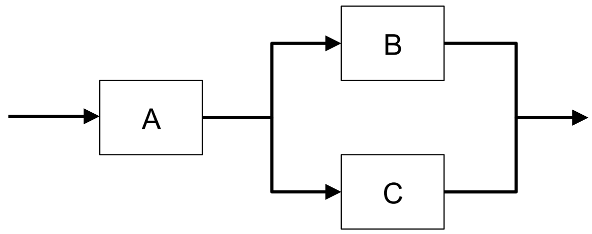


Figure 7: System considered for Examples 1 and 2.

6.1. Example 1: series/parallel configuration

The first example consists of 3 components arranged in a series/parallel configuration as shown in Fig. 19. In this case the following probabilities of failures (on-demand) are provided:

- $p_A = 1.0 \cdot 10^{-2}$
- $p_B = 5.0 \cdot 10^{-2}$
- $p_C = 1.0 \cdot 10^{-1}$

From a RISMC (i.e., dynamic PRA) point of view the analysis of this system is performed as follows (see Section 3.1):

- Define 3 stochastic parameters (i.e., $S = 3$):
  - s_1: status of component A
  - s_2: status of component B
  - s_3: status of component C
- Assign a distribution to each stochastic parameter; in this case a Bernoulli distribution
- Define I_i^+ and I_i^- for each distribution: in this case we have chosen $I_i^- = [0.0, 0.1]$ and $I_i^+ = [1.0, 1.1]$
- Generate N samples, for example by employing Monte-Carlo or Grid sampling strategies
- Determine R_0, R_i^- and R_i^+ for each component
- Determine the desired RIMs for each component

Note that a Monte-Carlo sampling is not the best sampling strategy in terms of computational costs. This is even more relevant if the value of p_A, p_B or p_C were several order of magnitude lower.

A more effective sampling strategy would be the Grid sampling (see Section 3.1): the stochastic variables are sampled over a fixed Cartesian grid and

Table 1: Results obtained for Example 1.

	Analytical	SAPHIRE	RAVEN
FV_A	0.6656	0.669	0.6656
FV_B	0.3311	0.34	0.3311
FV_C	0.3311	0.34	0.3311
	Analytical	SAPHIRE	RAVEN
RAW_A	66.889	66.9	66.88963211
RAW_B	7.2909	7.29	7.2909699
RAW_C	3.97993	3.98	3.97993311

a probability weight is associated to each sample. In this case, each stochastic variable s_i is sampled over two values, 0.0 and 1.0, and the probability weights w_i^0 and w_i^1 values associated to each sample coordinate are:

- $s_i = 0.0$: $w_i^0 = prob(s_i \in [-\infty, 0.5])$
- $s_i = 1.0$: $w_i^1 = prob(s_i \in [0.5, +\infty])$

Following this grid sampling strategy, only $2^3 = 8$ are needed. Below, the FV and RAW importance values for all three components obtained by RAVEN (using a Grid sampling strategy) are shown compared with the analytical ones.

6.2. Example 2: time-dependent stand-by configuration

The second example considers a simplified ECCS model (see Fig. **??**) of a reactor. It consists of the following components and for a subset of them a value of mean time to failure (MTTF) is provided:

- Motor-operate valve M (MTTF = 24 h, $\lambda_{valve} = 0.041667$)
- Two redundant pumps, pump1 and pump2 (MTTF = 12 h, $\lambda_{pump1} = \lambda_{pump2} = 0.083333$)
- Heat exchanger HX (reliability = 1.0)

Pump1 is normally used while pump2 is on standby. If Pump1 fails then pump2 provide water flow. Pump2 cannot fail while in standby. Switch from pump1 to pump2 is perfectly reliable. The cooling is such that it takes 2 hours to reach vessel failure condition if the M-pump1-pump2 system has failed. Mission time is again equal to 24 hours.

Note in this case classical PRA methods require few model simplifications in order to correctly determine system reliability. Below are shown the FV importance for all three components obtained by RAVEN (using a Monte-Carlo sampling strategy) compared with the analytical ones.

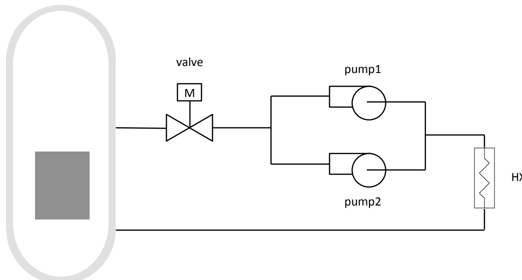


Figure 8: System considered for Example 3.

Table 2: Results obtained for Example 2.

	Analytical	SAPHIRE	RAVEN
FV_{valve}	0.301708135	0.697	0.303310397
FV_{pump1}	0.256887262	0.824	0.258927596
FV_{pump2}	0.256887262	0.824	0.258927596

	Analytical	SAPHIRE	RAVEN
RAW_{valve}	1.175587151	1.10	1.179402119
RAW_{pump1}	1.117058099	1.05	1.118857043
RAW_{pump2}	1.117058099	1.05	1.118857043

6.3. Example 3: K out of N configuration

The third example is similar to the one shown in Section 6.2 It consists of the following components:

- Motor-operate valve M (MTTF = 24 h, $\lambda_{valve} = 0.041667$)
- Three pumps, pump1, pump2 and pump3 (MTTF = 12 h, $\lambda_{pump1} = \lambda_{pump2} = \lambda_{pump3} = 0.083333$)
- Heat exchanger HX (reliability = 1.0)

All pumps are initially running but 2 out of 3 are required to cool the system. Mission time is again equal to 24 hours.

Below are shown the FV importance for all three components obtained by RAVEN (using a Monte-Carlo sampling strategy) compared with the analytical ones.

6.4. Example 4: time and physics dependent stand-by configuration

The forth example considers the system of Example 2 (see Section 6.2) but it considers also the temporal behavior of the reactor. The top event is not the failure of valve-pump1-pump2 system but it occurs when core temperature T reaches a threshold value T_{max} (i.e., reactor failure). The analysis is performed for two different power levels: 100% and 120%. Mission time is still 24 hours.

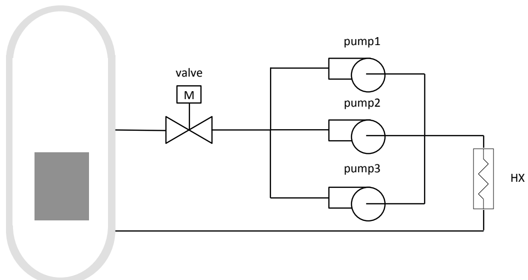


Figure 9: System considered for Example 3.

Table 3: Results obtained for Example III.

	Analytical	SAPHIRE	RAVEN
FV_{valve}	0.032191338	0.636	0.032625538
FV_{pump1}	0.075840848	0.942	0.080560827
FV_{pump2}	0.075840848	0.942	0.080560827
FV_{pump3}	0.075840848	0.942	0.080560827

	Analytical	SAPHIRE	RAVEN
RAW_{valve}	1.018734614	1.01	1.020401014
RAW_{pump1}	1.011870432	1.0	1.012910957
RAW_{pump2}	1.011870432	1.0	1.012910957
RAW_{pump3}	1.011870432	1.0	1.012910957

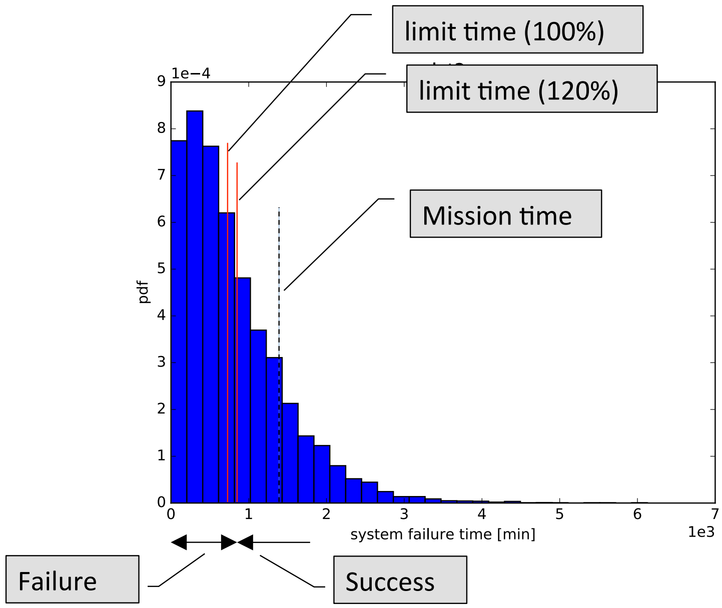


Figure 10

Table 4: Results obtained for Example 4.

	FV	RAW
valve		
pump1		
pump1		

Note that the configuration is slightly different from the one presented in the first two examples (here a stand-by configuration is introduced) but also the condition of system failure is dictated by the dynamic behavior of the PWR. The system is designed such that a late failure of the ECCS may not lead to system failure (i.e., natural circulation is providing enough cooling). In other words, the ECCS is vital especially in the hours right after a reactor scram.

In principle, this test case cannot be solved analytically due to the complexity of the reactor behavior. Practically it can be solve by identifying the time T_{lim} (limit time) after which a failure of the valve-pump1-pump2 system does not cause reactor failure. Note that T_{lim} could only be determined be recursively run the system simulator.

7. Extension To Time-Dependent Data

In order to extend the calculation presented above in the time-domain we need additional information: the temporal profile of the status of those components that might be taken offline due to maintenance or testing.

We will follow this notation:

- Ξ represents the system configuration, i.e., the status of components and systems of the plant on the time scale τ of the plant lifetime
- $RAW_i(\tau),\ FV_i(\tau),\ RRW_i(\tau)),\ B_i(\tau)$ are the RIMs determined for the basic event i calculated on the time scale τ

Note that in our application the status of each component can be only binary: component operating or component off-line (i.e., either because it is failed or under maintenance/testing). Thus component performance degradation is not considered. The calculation algorithms is as follows given a set of simulated data:

1. Divide the temporal profile into L segments where the status of the components, i.e. the system configuration Ξ, remain constant
2. For each time segment, i.e., for $l = 1, \ldots, L$

   2.1. Determine R_0 according to the system configuration Ξ_l for segment l

   $$R_0(l) = \frac{N_{CD,\Xi=\Xi_l}}{N_{\Xi=\Xi_l}} \tag{7}$$

   2.2. For each component determine R_i^- and R_i^+

   - If the component is on-line, R_i^- and R_i^+ are determined as follows:

     $$R_i^+(l) = \frac{N_{CD,s_i \in I_i^+,\Xi=\Xi_l}}{N_{\Xi=\Xi_l}} \tag{8}$$

     $$R_i^-(l) = \frac{N_{CD,s_i \in I_i^-,\Xi=\Xi_l}}{N_{\Xi=\Xi_l}} \tag{9}$$

   - If the component is off-line determine $R_i^+(l)$ according to Eq. **??** and set $R_i^-(l) = R_i^+(l)$

8. Test example

For the scope of this paper we have chosen an elementary example that can help the reader to understand the proposed algorithm. This a simple system composed of three components (i.e., A, B and C) in a parallel/series configuration. To each component a failure rate is provided when the system is called on demand:

- $\lambda_A = 1.0 \cdot 10^{-3} hr^{-1}$
- $\lambda_B = 5.0 \cdot 10^{-3} hr^{-1}$
- $\lambda_C = 1.0 \cdot 10^{-2} hr^{-1}$

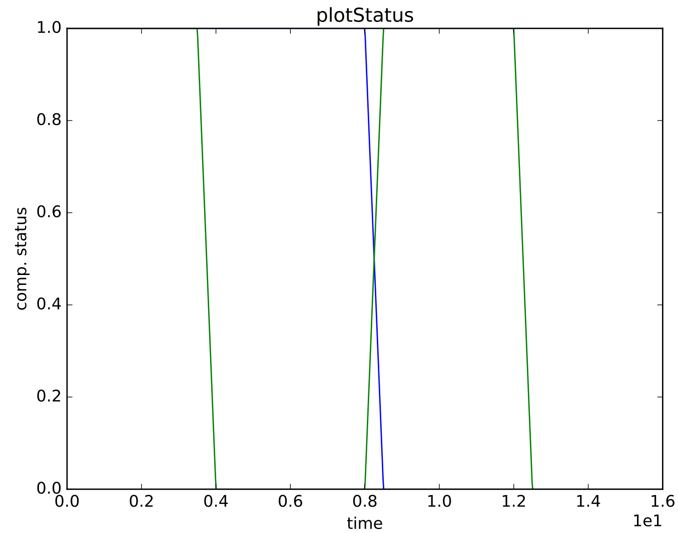


Figure 11: Temporal profile of the status for component A (blue line) and C (green line).

Even though the system can be solved analytically we have chosen a dynamic method to solve it in order to show how the proposed methods is implemented. By using a Monte-Carlo based Dynamic PRA method, we have generated a database of simulated data where each data point is structured as follows:

- Input variables: failure time of components A, B and C (i.e., s_i) sampled from their own distribution (i.e., exponential with lambda values provided at the beginning of this section)
- Output variables: status of the system (either OK or CD)

We have selected the temporal profile for two components: A and C as shown in Fig. 11. By following the algorithm presented above it has been possible to determine the temporal profiles of:

- system failure probability (see Fig. 12)
- RIMs such as FV (see Fig. 13)

9. New RIMs in a Dynamic PRA Context

Note that the RIMs described so far are tight to a binary logic of the outcome variable (e.g., OK vs. CD). Dynamic PRA approaches typically generate a continuous value of the outcome variables (e.g., peak clad temperature - PCT). In our application (see previous sections) we typically convert PCT to a discrete one as follows:

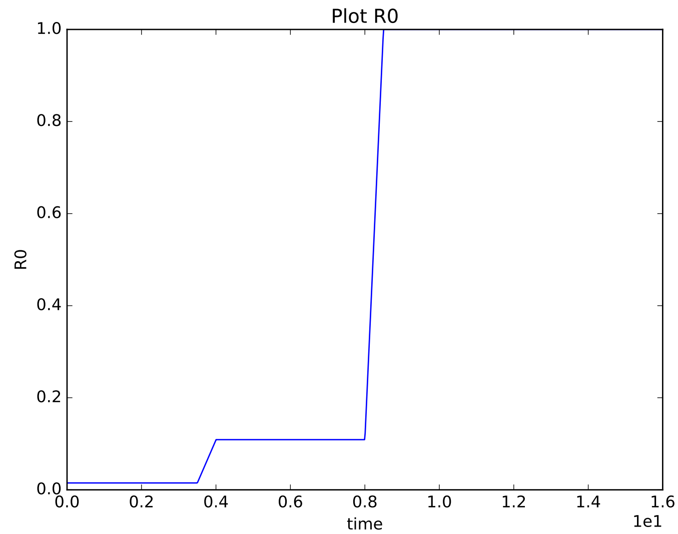


Figure 12: Temporal profile for system failure probability.

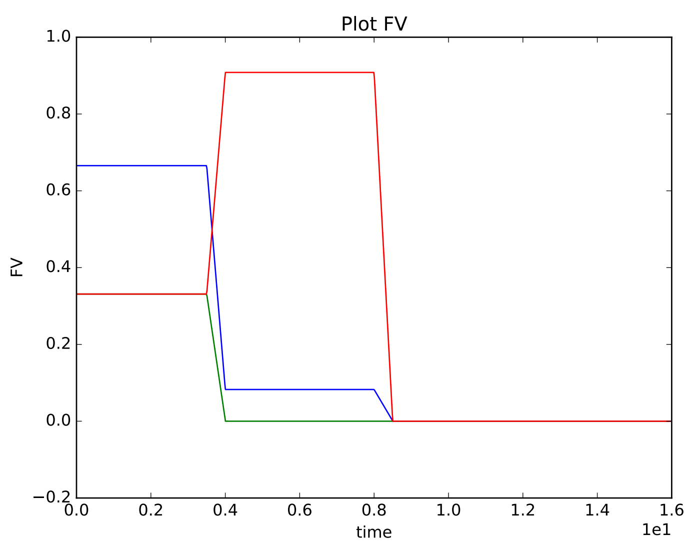


Figure 13: FV profile for components B (green line), A (blue line) and C (red line).

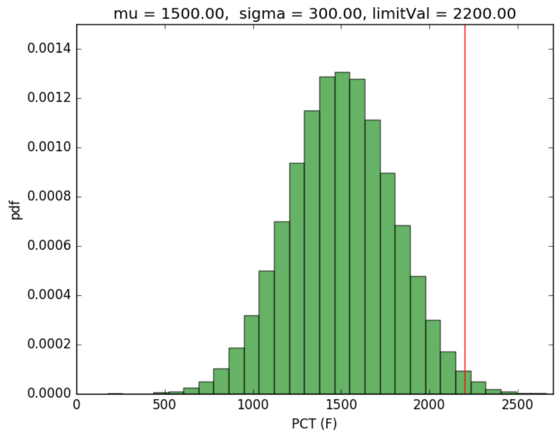


Figure 14: Plot of a hypothetical $pdf_0(T)$

- $PCT > 2200F$: outcome = CD
- $PCT < 2200F$: outcome = OK

Given the different structure of the approach used in this paper to solve a PRA problem (i.e., Dynamic instead of classical PRA), the reader might think that a different set of RIMs should/could be developed in order to capture the nature of the problem solved using Dynamic PRA. As a starting point, it would worth investigating the nominal probabilistic distribution (pdf) of PCT with the one obtained when reliability of each basic event (sampled parameter) is 0.0 or 1.0. So now we can indicate:

- $pdf_0(T)$: nominal pdf of PCT
- $pdf_i^-(T)$: pdf of PCT associated to basic event i assuming basic event is perfectly reliable
- $pdf_i^+(T)$: pdf of PCT associated to basic event i assuming basic event has failed

An example is shown below for a hypothetical case where obtained $pdf_0(T)$ is indicated using an histogram while the limit value for PCT is shown using the red line passing at 2200 F. In order to make a connection to what has been presented in the previous section, note that by looking at Fig. 14:

$$R_0 = \int_{2200}^{\infty} pdf_0(T)\, dT \tag{10}$$

As part of the RISMC analysis, the user might want to supplement the results obtained in the previous section with the information associated to a

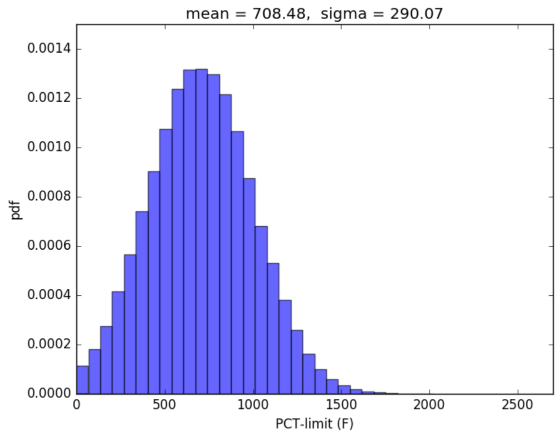


Figure 15: Plot of $margin_0$ for the case shown in Fig. 14

more effective margin analysis. In particular, of interest for RISMC applications is (see Fig. 15) the concept of margin:

$$margin = 2200 - PCT \text{ given } PCT < 2200 \tag{11}$$

Using the same philosophy indicated in the previous section for classical RIMs, we want to determine:

- $margin_0$: pdf of the variable $2200 - PCT$ given that $PCT < 2200$
- $margin_i^-$: pdf of the variable $2200 - PCT$ given that $PCT < 2200$ for basic event i assuming it is perfectly reliable
- $margin_i^+$: pdf of the variable $2200 - PCT$ given that $PCT < 2200$ for basic event i when its assumed to be failed

Note now that $margin_0$, $margin_i^-$ and $margin_i^+$ are now pdfs and not numerical values. Hence, now the challenge arises on how to compare two pdfs:

- $margin_0$ vs. $margin_i^-$
- $margin_0$ vs. $margin_i^+$

Assume two pdfs are given: $pdf_1(x)$ and $pdf_2(x)$. Few approaches can be followed: Z-test or KolmogorovSmirnov test. In the first approach (Z-test), the following variable Z is computed:

$$Z_{1,2} = \frac{mean(pdf_1) - mean(pdf_2)}{\sqrt{std\_dev^2(pdf_1) - std\_dev^2(pdf_2)}} \tag{12}$$

where:

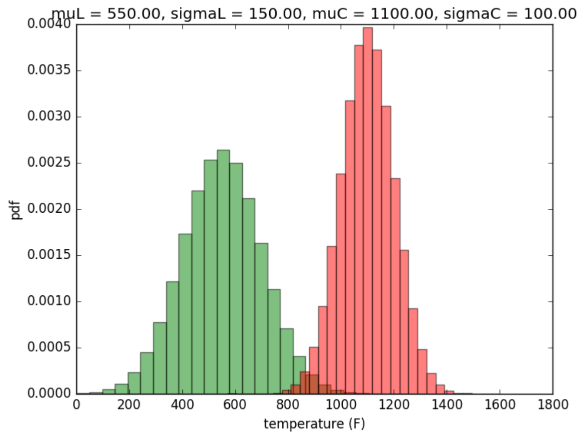


Figure 16: Plot of the pdfs for PCT (green) and CFT (red).

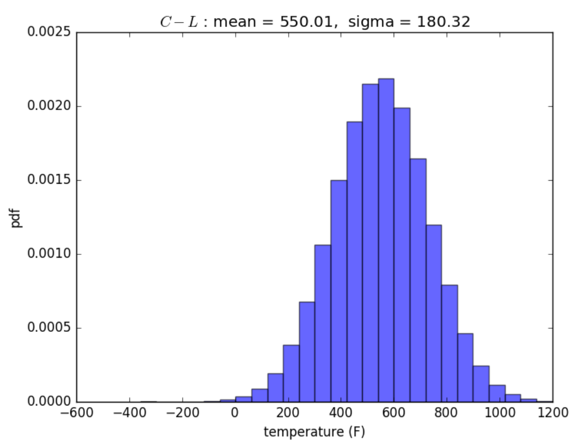


Figure 17: Plot of the pdf of the variable $CFT - PCT$.

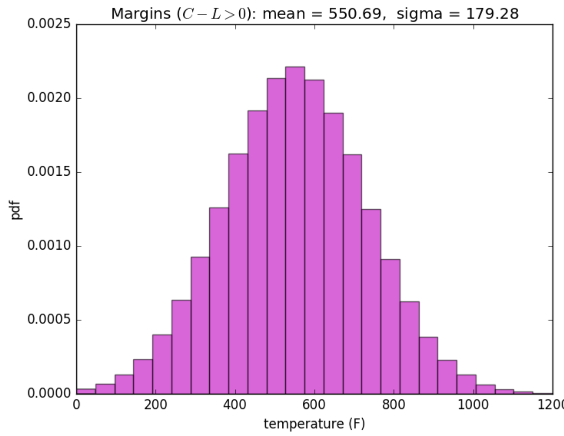


Figure 18: Plot of the pdf of the margin, i.e., $CFT - PCT > 0$.

- $mean(pdf)$ correspond to the mean of $pdf(x)$
- $std\_dev(pdf)$ correspond to the standard deviation of $pdf(x)$

In the second approach (KolmogorovSmirnov test []), instead of the pdf, the cumulative distribution functions (pdf) are considered: $cdf_1(x)$ and $cdf_2(x)$. In particular, the Kolmogorov-Smirnov statistic is calculated as:

$$Z_{1,2} = \sup_x (cdf_1(x) - cdf_2(x)) \tag{13}$$

Note that so far we have imposed clad failure temperature (CFT) to be a fixed value, i.e., 2200 F. In many RISMC applications CFT is no longer a numerical value but it can be un uncertain parameter, i.e., a pdf is associated to CDF: $pdf(T)$. This link goes back to the original logo of RISMC where a pdf for "load" and "capacity" (see Fig. **??**). A new definition of margin can be then defined:

$$margin = (CFT - PCT) \text{ given } (CFT - PCT > 0) \tag{14}$$

From here, once the pdf associated to the margin variable is determined it is possible to employ either the Z-tests or the KolmogorovSmirnov test in order to measure how this pdf changes when each basic event is considered perfectly reliable or failed.

10. Conclusions

This paper has presented a mathematical framework for determining risk importance measures in a simulation based, i.e. dynamic, PRA framework. We have shown how classical measures can be derived and we have provided few explanatory examples. We have also indicated how the data generation method is extremely important to maximize the amount of information generated by each simulation run. Lastly we have presented an additional set of risk importance measures that are not bounded by a Boolean logic but explore the continuity of the problem. The advantage of these measures is that they capture the idea of "safety margin".

Appendix A: Analytical Results

Example 1

For the system described in Section 6.1 we have the following

$$R_0 = p_A + p_B p_C - p_A p_B p_C = 0.01495 \tag{15}$$
$$R_A^- = p_B p_C = 0.005 \tag{16}$$
$$R_A^+ = 1.0 \tag{17}$$
$$R_B^- = p_A = 0.01 \tag{18}$$
$$R_B^+ = p_A + p_C - p_A p_C = 0.109 \tag{19}$$
$$R_C^- = p_A = 0.01 \tag{20}$$
$$R_C^+ = p_A + p_B - p_A p_B = 0.0595 \tag{21}$$

Thus:

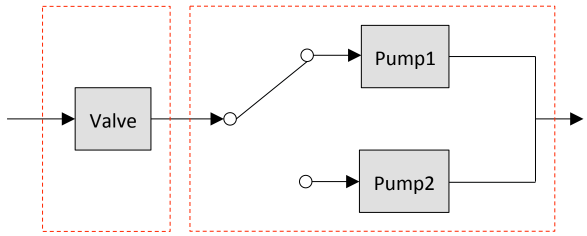
$$FV_A = \frac{R_0 - R_A^-}{R_0} = 0.665552 \tag{22}$$
$$FV_B = \frac{R_0 - R_B^-}{R_0} = 0.331104 \tag{23}$$
$$FV_C = \frac{R_0 - R_C^-}{R_0} = 0.331104 \tag{24}$$

and:

$$RAW_A = \frac{R_i^+}{R_0} = 66.88963 \tag{25}$$
$$RAW_B = \frac{R_i^+}{R_0} = 7.29097 \tag{26}$$
$$RAW_C = \frac{R_i^+}{R_0} = 3.97993 \tag{27}$$

Example 2

We can solve the system described in Section 6.2 using reliability block diagrams (see Fig. 19).



$$R_{sys}(t) = R_{valve}(t) \cdot \int_0^t f_{pump1}(\tau) \cdot R_{pump2}(t-\tau)d\tau$$

Figure 19: Reliability block diagrams for Example 2 (see Section 6.2).

Thus time dependent reliability of the system $R_sys(t)$ as function of time t:

$$R_{sys}(t) = R_{valve}(t) \int_0^t f_{pump1}(\tau) R_{pump2}(t-\tau)d\tau \tag{28}$$

where:

- $R_{valve}(t) = e^{-\lambda_{valve} t}$
- $f_{pump1}(t) = \lambda_{pump1} e^{-\lambda_1 t}$
- $R_{pump2}(t) = e^{-\lambda_{pump2} t}$

It can be shown that if $\lambda_{pump1} = \lambda_{pump2} = \bar{\lambda}$:

$$R_{sys}(t) = e^{-\lambda_{valve} t}[e^{-\bar{\lambda}}(1 + \bar{\lambda} t)] \tag{29}$$

For this system we have the following for a mission time $T = 24$ hours:

$$R_0 = 1.0 - R_{sys}(T) = 0.85063876 \tag{30}$$
$$R^-_{valve} = 1.0 - [e^{-\bar{\lambda} T}(1 + \bar{\lambda} T)] = 0.593994129 \tag{31}$$
$$R^+_{valve} = 1.0 \tag{32}$$
$$R^-_{pump1} = 1.0 - R_{valve}(T) = 0.6321205 \tag{33}$$
$$R^+_{pump1} = 1.0 - R_{valve}(T) R_{pump2}(T) = 0.95021292 \tag{34}$$
$$R^-_{pump2} = R^-_{pump1} = 0.6321205 \tag{35}$$
$$R^+_{pump2} = R^+_{pump1} = 0.95021292 \tag{36}$$

Example 3

We can solve the system described in Section 6.2 using again reliability block diagrams. Thus time dependent reliability of the system $R_sys(t)$ as function of time t:

$$R_{sys}(t) = R_{valve}(t) R_{2oo3}(t) \tag{37}$$

where $R_{2oo3}(t)$ represents reliability of a set of three identical components in a 2 out of 3 (2oo3) configuration.

It can be shown that if $\lambda_{pump1} = \lambda_{pump2} = \lambda_{pump3} = \bar{\lambda}$ (thus $R_{pump1}(t) = R_{pump2}(t) = R_{pump3}(t) = \bar{R}(t) = e^{-\bar{\lambda}t}$) then $R_{2oo3}(t)$ can be written as

$$R_{2oo3}(t) = \sum_{n=2}^{3} \binom{3}{n} \bar{R}(t)^n [1 - \bar{R}(t)]^{3-n} = 3e^{-2\bar{\lambda}t}(1 - e^{-\bar{\lambda}t}) + e^{-3\bar{\lambda}t} \tag{38}$$

For this system we have the following for a mission time $T = 24$ hours:

$$R_0 = 1.0 - R_{sys}(T) = 0.981609917 \tag{39}$$
$$R^-_{valve} = 1.0 - R_{2oo3}(T) = 0.95001058 \tag{40}$$
$$R^+_{valve} = 1.0 \tag{41}$$
$$R^-_{pump1} = 1.0 - R_{1oo2}(T)R_{valve}(T) = 0.907163789 \tag{42}$$
$$R^+_{pump1} = 1.0 - R_{valve}(T)\bar{R}(t)^2 = 0.993262051 \tag{43}$$
$$R^-_{pump2} = R^-_{pump1} \tag{44}$$
$$R^+_{pump2} = R^+_{pump1} \tag{45}$$
$$R^-_{pump3} = R^-_{pump1} \tag{46}$$
$$R^+_{pump3} = R^+_{pump1} \tag{47}$$

where $R_{1oo2}(t)$ represents reliability of a set of two identical components in a 1 out of 2 (1oo2) configuration:

$$R_{1oo2}(t) = \sum_{n=1}^{2} \binom{2}{n} \bar{R}(t)^n [1 - \bar{R}(t)]^{2-n} = 2e^{-\bar{\lambda}t}(1 - e^{-\bar{\lambda}t}) + e^{-2\bar{\lambda}t} \tag{48}$$

References

[1] W. E. Vesely, T. C. Davis, R. S. Denning, N. Saltos, NUREC/CR-3385: Measures of Risk Importance and their Applications, Technical Report, U.S. Nuclear Regulatory Commission, Washington DC, 2005.

[2] K. Fleming, Developing useful insights and avoiding misleading conclusions from risk importance measures in PSA application, in: Proceedings of PSA 1996 International Topical Meeting On Probabilistic Safety Assessment And Analysis, 1996.

[3] U. S. NRC, NUREC/CR-1150: Severe Accident Risks: An Assessment For Five U.S. Nuclear Power Plants Final Summary Report, Technical Report, U.S. Nuclear Regulatory Commission, Washington DC, 2005.

[4] J. Devooght, C. Smidts, Probabilistic dynamics as a tool for dynamic PSA, Reliability Engineering & System Safety 52 (1996) 185 – 196.

[5] C. Rabiti, A. Alfonsi, J. Cogliati, D. Mandelli, R. Kinoshita, RAVEN, a new software for dynamic risk analysis, in: Proceedings of the Probabilistic Safety Assessment andnd Management (PSAM) 12, 2014.

[6] B. Rutt, U. Catalyurek, A. Hakobyan, K. Metzroth, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, Distributed dynamic event tree generation for reliability and risk assessment, in: 2006 IEEE Challenges of Large Applications in Distributed Environments, 2006, pp. 61–70.

[7] K. S. Hsueh, A. Mosleh, The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants, Reliability Engineering & System Safety 52 (1996) 297–314.

[8] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, M. Woltereck, An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties, Reliability Engineering & System Safety 77 (2002) 229–238.

[9] R. C. D. Team, RELAP5-3D Code Manual, Technical Report, Idaho National Laboratory Technical Report, 2005.

[10] R. O. Gauntt, MELCOR Computer Code Manual, Version 1.8.5, Vol. 2, Rev. 2, Sandia National Laboratories, NUREG/CR-6119, 2000.

[11] EPRI, Modular Accident Analysis Program 5 (MAAP5) Applications Guidance: Desktop Reference for Using MAAP5 Software - Phase 2 Report, Technical Report, Electric Power Reserach Institute, Palo Alto (CA), 2013.

[12] C. Smith, C. Rabiti, R. Martineau, Risk Informed Safety Margins Characterization (RISMC) Pathway Technical Program Plan, Technical Report, Idaho National Laboratory Technical Report: INL/EXT-11-22977, 2011.

[13] D. Mandelli, C. Smith, T. Riley, J. Nielsen, A. Alfonsi, J. Cogliati, C. Rabiti, J. Schroeder, BWR station blackout: A RISMC analysis using RAVEN and RELAP5-3D, Nuclear Technology 193 (2016) 161–174.

[14] D. Maljovec, S. Liu, B. Wang, D. Mandelli, P. T. Bremer, V. Pascucci, C. Smith, Analyzing simulation-based PRA data through traditional and topological clustering: A BWR station blackout case study, Reliability Engineering & System Safety 145 (2015) 262–276.

[15] D. Mandelli, S. Prescott, C. Smith, A. Alfonsi, C. Rabiti, J. Cogliati, R. Kinoshita, Modeling of a flooding induced station blackout for a pressurized water reactor using the RISMC toolkit, in: Proceedings of PSA 2015 International Topical Meeting On Probabilistic Safety Assessment And Analysis, 2015.

[16] R. L. Boring, R. B. Shirley, J. C. Joe, D. Mandelli, C. Smith, Simulation And Non-Simulation Based Human Reliability Analysis Approaches, Technical Report, Idaho National Laboratory Technical Report: INL/EXT-14-33903, 2014.

[17] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, R. Kinoshita, A. Naviglio, RAVEN and dynamic probabilistic risk assessment: Software overview, in: Proceedings of European Safety and Reliability Conference (ESREL 2014), Wroklaw (Poland), 2014.

[18] E. Zio, M. Marseguerra, J. Devooght, P. Labeau, A concept paper on dynamic reliability via monte carlo simulation, Mathematics And Computers In Simulation 47 (1998) 371–382.

[19] J. C. Helton, F. J. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliability Engineering & System Safety 81 (2003).

[20] A. Amendola, G. Reina, DYLAM-1, a software package for event sequence and consequence spectrum methodology, in: EUR-924, CEC-JRC. ISPRA: Commission of the European Communities, 1984.

[21] G. Cojazzi, The DYLAM approach for the dynamic reliability analysis of systems, Reliability Engineering and System Safety 52 (1996) 279–296.

[22] C. Rabiti, D. Mandelli, A. Alfonsi, J. Cogliati, R. Kinoshita, Introduction of supervised learning capabilities of the RAVEN code for limit surface analysis, in: Proceedings of American Nuclear Society (ANS), Reno (NV), 2014.

[23] D. Mandelli, C. Smith, Adaptive sampling using support vector machines, in: Proceeding of American Nuclear Society (ANS), San Diego (CA), volume 107, 2012, pp. 736–738.

[24] H. S. Abdel-Khalik, Y. Bang, J. M. Hite, C. B. Kennedy, C. Wang, Reduced order modeling for nonlinear multi-component models, International Journal On Uncertainty Quantification (2012) 341–36.

[25] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, R. Kinoshita, Adaptive dynamic event tree in RAVEN code, in: Proceedings of American Nuclear Society (ANS), Anaheim (CA), 2014.

[26] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, R. Kinoshita, A. Naviglio, Dynamic event tree analysis through RAVEN, in: Proceedings of PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis Columbia, SC, on CD-ROM, American Nuclear Society, LaGrange Park, IL, 2013.