INL/EXT-20-59928

Light Water Reactor Sustainability Program

Integration of Data Analytics with Plant System Health Program

September 2020

U.S. Department of Energy

Office of Nuclear Energy

INL/EXT-20-59928

Integration of Data Analytics with Plant System Health Program

D. Mandelli, C. Wang, J. Cogliati, C. Smith (Idaho National Laboratory)

S. Hess, R. Sugrue (Jensen Hughes)

C. Pope, J. Miller, S. Ercanbrack (Idaho State University)

D. Cole, J. Yurko (University of Pittsburgh)

September 2020

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy

EXECUTIVE SUMMARY

This report summarizes the research and development R&D activities of the Plant Health Management (PHM) project during fiscal year 2020 (FY-20). This project focuses on the development of methods that integrate component health data and propagate such information at the system level to evaluate system sources of risk (where herein the term risk is broadly construed to include both plant safety and economics). Project development lives in cooperation with the Risk Informed Asset Management (RIAM) project. In fact, the RIAM project, uses the reliability models and data generated by the PHM project to optimize plant operations (e.g., maintenance and replacement schedule, optimal maintenance posture, etc.). This year's activities for the PHM project focused mainly on the development of two classes of models. The first class includes a series of component reliability models which include aging, testing and maintenance. The second class includes a series of system reliability models which focus on the secondary side (i.e., power conversion portion) of exiting U.S. reactors. These models are based on fault tree logic structures such that they can be used within the framework of the existing plant Probabilistic Risk Assessment (PRA) methods and software. Our team also started to focus on the management of health data which was tackled in two directions. The first one focuses on the integration of monitored plant data and simulation models to assess component health. This approach moves from a classical data based to a model+data based approach with the goal of providing a more comprehensive and predictive framework for use of component health information. The second direction focuses on linking equipment reliability data (e.g., maintenance/failure reports, component monitoring data, etc.) directly to system reliability models using a margin based framework rather than one that is probability based. The main advantage of a margin based approach is that it can provide to responsible plant engineers and their management (decision-makers) with a more tangible and comprehensive set of information on system/component health and to predict how performance is likely to change in the future. Such predictive capability will allow for safer and more cost-effective plant performance in both the short-term (which is the focus of PHM) and long-term (the focus of RIAM).

CONTENTS

FIGURES

TABLES

ACRONYMS

API	Application Programming Interface
ASME	American Society for Mechanical Engineers
BE	Basic Event
BWR	Boiling Water Reactor
CBM	Condition Based Maintenance
CDF	Core Damage Frequency
CDF	(Probabilistic) Cumulative Distribution Function
CM	Corrective Maintenance
DOE	(United States) Department of Energy
EER	Economic Enterprise Risk
EPRI	Electric Power Research Institute
ER	Equipment Reliability
ERWG	Equipment Reliability Working Group
ET	Event Tree
FEG	Functional Equipment Group
FMEA	Failure Modes and Effects Analysis
FT	Fault Tree
GADS	Generating Availability Data System
GPM	Gaussian Process Model
GRA	Generation Risk Assessment
HPCI	High Pressure Coolant Injection
INL	Idaho National Laboratory
INPO	Institute of Nuclear Power Operations
IR	Issue Report
IRR	Internal Rate of Return
JH	Jensen Hughes
LWR	Light Water Reactor
LWRS	Light Water Reactor Sustainability
M&D	Monitoring and Diagnostic
MCMC	Markov Chain Monte Carlo
MCS	Minimal Cut Set

MTF	Margin to Failure
MTBF	Meant Time Between Failure
MTTF	Meant Time to Failure
MTTR	Meant Time to Repair
NERC	North American Electric Reliability Corporation
NEUP	Nuclear Energy University Program
NMTF	Normalized Margin to Failure
NPP	Nuclear Power Plant
NPV	Net Present Value
NRC	Nuclear Regulatory Commission
O&M	Operation and Maintenance
PCA	Principal Component Analysis
PDF	Probability Density Function
PdM	Predictive Maintenance
PHM	Plant Health Management
PM	Preventive Maintenance
PMMD	Proactive Management of Materials Degradation
PRA	Probabilistic Risk Assessment
PSH	Plant System Health
PWR	Pressurized Water Reactor
RAM	Reliability, Availability, and Maintainability
RAVEN	Risk Analysis Virtual Environment
RCM	Reliability Centered Maintenance
RI	Risk-Informed
RIAM	Risk-Informed Asset Management
RIDM	Risk-Informed Decision-Making
RISA	Risk Informed Systems Analysis
RI-PSH	Risk Informed Plant System Health
RMTS	Risk-Managed Technical Specifications
ROM	Reduced Order Model
RTF	Run to Failure
RTM	Run to Maintenance
RUL	Remaining Useful Life
R&D	Research and Development

SFCP	Surveillance Frequency Control Program
SRCM	Streamlined Reliability Centered Maintenance
SR^2ML	Safety, Risk, Reliability Model Library
SSCs	Structures, Systems, and Components
STI	Surveillance Test Intervals
TE	Top Event
TS	Technical Specifications
T&M	Testing and Maintenance
UQ	Uncertainty Quantification
VBM	Value Based Maintenance
VERT	Versatile Economic Risk Tool
WO	Work Orders

Integration of Data Analytics with Plant System Health Program

1. INTRODUCTION

This report summarizes the R&D activities of the Plant Health Management (PHM) project during fiscal year 2020 (FY-20). This project focuses on the development of methods that integrate component health data and propagate this information at the system level to evaluate system sources of risk. Note that for the PHM project, term risk is broadly construed. In addition to plant safety (such as determined in a plant Probabilistic Risk Assessment [PRA]), risk also includes items that can impact plant operational and economic performance. From the viewpoint of the Light Water Reactor Sustainability (LWRS) program, both aspects (safety and economics) are necessary conditions that the fleet of operating Nuclear Power Plants (NPPs) need to meet.

Sources of component health data can be testing data, maintenance/failure reports, or monitoring streaming data (such as physical process parameters like system pressure, flow, etc.). Typically, this kind of information is stored in the plant Equipment Reliability (ER) database and/or in the Remote Monitoring and Diagnostic (M&D) Center.

This project development lives in symbiosis with the Risk Informed Asset Management (RIAM) project. The RIAM project, in fact, uses the reliability models and data generated by the PHM project to optimize plant operations (e.g., maintenance/replacement schedule, optimal maintenance posture, etc.). The RIAM project explicitly addresses plant decision making by framing each of these decisions in an optimization form with constraints set by the user. The primary difference between the PHM and RIAM efforts is timeframe. The focus on PHM is on short to medium term (i.e., up to 2 operating cycles) whereas the focus of RIAM is longer term (i.e., to the projected end of plant life – typically to the end of first (60 years) or second license renewal (80 years) in the United States).

This year's activities for the PHM project focused mainly on the development of two classes of models. The first one includes a series of component reliability models which include aging, testing and maintenance. The second class includes a series of system reliability models which focus on the secondary side (i.e., power conversion portion) of exiting U.S. reactors. These models are based on Fault Tree (FT) logic structures such that they can be applied using existing PRA methods and software. While existing plant PRAs typically focus mainly on plant safety systems, this second class of models consider the generation side of the plant. Typically, these models are known as Generation Risk Assessment (GRA) models (see Section 5). GRA models explicitly measure economic risk (e.g., loss of power generation) rather than a safety measure of risk (as it is done in a plant PRA).

We also started to focus on the management of health data associated with plant structures, systems, and components (SSCs) and we tackled this issue in two directions. The first focuses on the integration of monitoring data with simulation models to assess component health (see Section 10). This approach is moving from a classical data-based approach to one which uses a model+data based approach with the goal of improving higher component health information. There exists a substantial literature focuses on using machine learning methods to predict component health solely based on past data [11]. This is a valid course of action for those applications where failure mechanism dynamics are not yet well understood. However, in several applications, computational models have been developed to understand and predict component

behavior and dynamics [24-30]. In this respect, these computational models can provide a valuable source of knowledge that can be employed to determine component health when coupled with monitoring data.

The second direction focuses on linking ER data (e.g., maintenance/failure reports, component monitoring data, etc.) directly to system reliability models using two kind of languages: a probability-based and a margin-based language (see Section 5). The main advantage of a margin-based language is that it can provide more tangible information on system/component health and how it propagates to the system level and over time to responsible plant system engineers and management decision-makers. In this research we show how these two languages compare when provided the same source of data.

In this report we also provide an exhaustive overview of modeling techniques. These techniques show how to link models together and how to integrate reliability with economic modeling. We show how these techniques have been developed within the RAVEN statistical framework and its associated plug-ins. Lastly, we summarize how the development of these models have been structured with the vision of releasing them to the industry and open-source tools.

1.1 Synergies Between RIAM and PHM Projects

The PHM and RIAM research are interrelated with the primary difference between the two being timeframe. The focus on PHM is on short to medium term (i.e., up to 2 operating cycles) whereas the focus of RIAM is longer term (i.e., to the projected end of plant life – typically to the end of first (60 years) or second license renewal (80 years) in the United States).

The PHM project is focusing on the development of models (e.g., component and system reliability models) and methods to integrate data (e.g., data generated by plant ER databases) into these models. The RIAM project is focusing on the development of methods designed to optimize plant resources (e.g., SSC, personnel, ER activities). The methods being developed and demonstrated for RIAM can be classified into two classes: sampling methods and optimization methods. Sampling methods are designed mainly to propagate data and model uncertainties (e.g., investment evaluation). Optimization methods are designed to determine the best solution to a problem that satisfies a limited set of criteria (also known as constraints).

Depending on the problem to be solved and the type of data available, the algorithm to be used might change. As an example, the data structure might be either discrete or continuous in nature. In addition, the problem under consideration might require a specific data set or, alternatively, a specific model which changes the problem structure depending on the considered boundary conditions.

Figure 1 provides a graphical look at the available algorithms developed within the RIAM project for plant resource management depending on the data structure and the method that the problem requires. Note that some of these methods have been developed during FY-19 (and they will be briefly presented) while several others have been developed during FY-20 (and they will be extensively described).

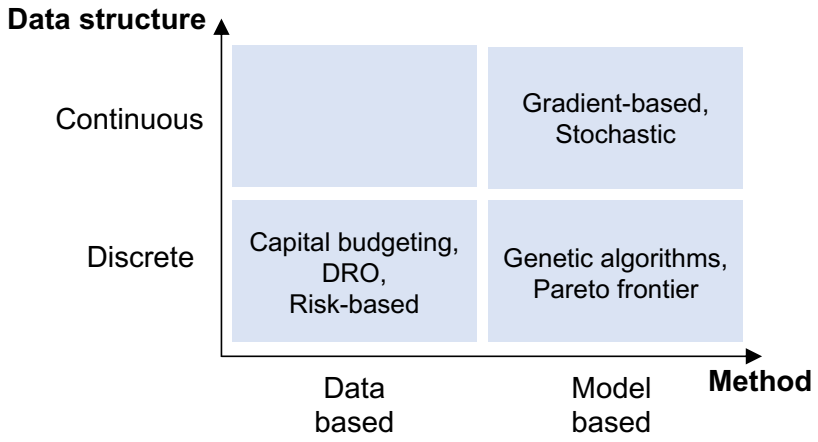


Figure 1. Classification of the optimization algorithms developed under the RIAM project based on employed data structured and the method being used.

2. COST AND RISK CATEGORIZATION RD&D PATH

The goal of the cost and risk categorization research path under the Risk Informed Systems Analysis (RISA) pathway for the LWRS program is to leverage advanced computational capabilities to support enhanced system performance and health management. The first objective of this effort is to integrate various elements of system health monitoring, management, and reporting in a manner that is significantly less labor intensive and is at least as technically effective as current programs. The second objective is to manage equipment and system performance and its impacts on financial risk and to support reduced costs associated with monitoring and regulatory compliance.

While the first objective is addressed by the PHM project, the second objective is addressed by the RIAM project. These two projects are coordinated to materialize the goal of the cost and risk categorization research path into a software platform that is referred to hereafter as the Risk Informed Plant System Health (RI-PSH) platform as shown in Figure 2.

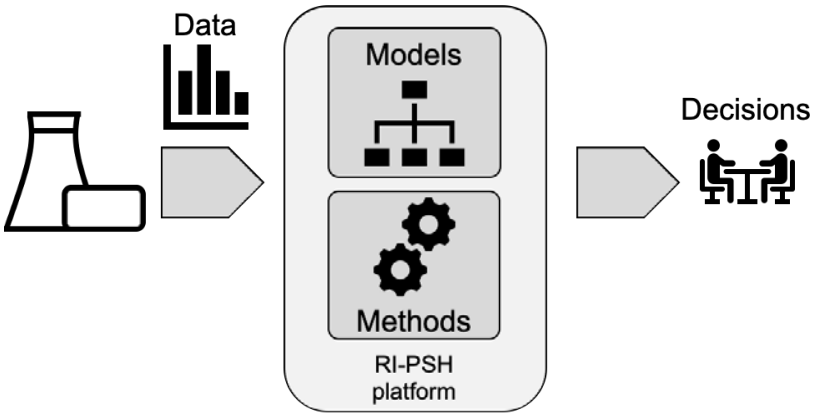


Figure 2. High level description of the RI-PSH platform.

The RI-PSH platform can be summarized as a Model Base System Engineering[a] (MBSE) [21] platform for system operations: from system health data, it provides decision-making knowledge on the best maintenance posture and optimal component maintenance/replacement schedule. By maintenance posture we are referring to the maintenance strategy for each component that reduces (and in an ideal case, minimizes) O&M costs while maintaining adequate system performance, reliability, and availability. The term MBSE summarizes the main feature of the RI-PSH platform: rather than focusing on specific O&M applications, we are providing a set of models and methods to the analyst, along with a computational analysis framework. Depending on the issue to be analyzed, the analyst can then assemble models together (e.g., a combination of reliability and cost models), and, apply a series of computational methods (e.g., optimization, uncertainty quantification, data analysis) to achieve desired outcomes.

From a development perspective, the PHM project is focusing mainly on the development of models and health data integration while the RIAM project is focusing on the optimization methods development. This development is coordinated with other projects with another LWRS pathway (e.g., the Plant Modernization pathway) and other U.S. Department of Energy–Nuclear Energy University Program (DOE–NEUP) projects. Figure 3 shows a complete picture of the external collaborators that contribute effectively to the PHM-RIAM projects. These collaborators include both academic and industry partners.

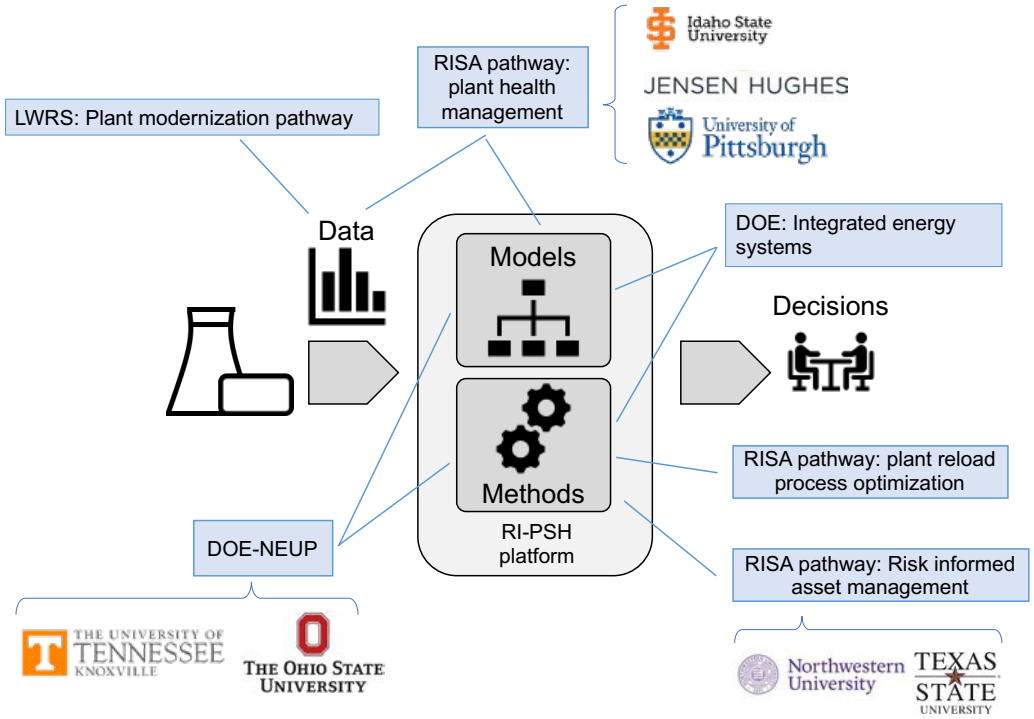


Figure 3. Interactions among the PHM project and other DOE projects in the development of the RI-PSH platform.

[a] MBSE is an emerging approach in the discipline of System Engineering (SE) which can be described as "the formalized application of modeling principles, methods, languages and tools to the entire lifecycle of large complex, interdisciplinary, sociotechnical systems" [151]. Compared to classical SE approaches which are based on the system-as-machine paradigm, MBSE is based on system-as-organism paradigm. In other words, MBSE evolves SE approaches to explicitly consider large highly complex, adaptive and human-interactive systems.

We have structured the RI-PSH platform on different repositories where each repository contains a specific class of models or methods. Figure 4 shows the hierarchical structure of the repositories currently being developed and maintained. The ones shown in yellow represent the ones fully developed under the RIAM-PHM projects while the ones shown in red are co-developed with other programs. A more detailed description of these repositories is presented as follows:

- **RAVEN [22]:** It is a flexible and multi-purpose modeling and simulation platform that supports Uncertainty Quantification (UQ), regression analysis, PRA, data analysis, and model optimization software. Depending on the tasks to be accomplished and on the probabilistic characterization of the problem, RAVEN perturbs the response of the system by altering its parameters through Monte-Carlo, latin hypercube, and other reliability surface search [8] sampling methods. The data generated by the sampling process are analyzed using classical 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 to perform UQ, reliability analysis (e.g., limit state surface), and parametric studies.
- **LOGOS:** This software contains a set of discrete optimization models that can be employed for capital budgeting optimization problems. LOGOS integrates economic and reliability risk in a single analysis framework. More specifically, provided SSC health information (e.g., failure rate or failure probability), O&M costs, replacement costs, cost associated with component failure, and plant budget constraints, LOGOS provides the optimal set of projects (e.g., SSC replacement or refurbishment) that maximizes profit and satisfies the provided requirements. The input data listed above can be either deterministic or stochastic in nature, i.e., they can be point values or probability distribution functions. In the latter case, several scenarios are generated by sampling of the provided distributions. The developed models are based on different versions of the knapsack optimization problem. Two main classes of optimization models have been initially developed: deterministic and stochastic. Stochastic optimization models evolve deterministic models by explicitly considering data uncertainties (associated to constraints or item cost and reward). These models can be employed as stand-alone models or interfaced with the INL developed RAVEN code to propagate data uncertainties and analyze the generated data (i.e., sensitivity analysis).
- **SR^{2}ML:** The Safety, Risk, Reliability Model Library (SR^{2}ML) is a software package that contains a set of reliability models designed to be interfaced with the INL developed RAVEN code. These models can be employed to perform both static and dynamic system risk analysis and determine risk importance of specific elements of the considered system. Two classes of reliability models have been developed; the first class includes all classical reliability models (FTs, ETs, Markov models and reliability block diagrams) which have been extended to deal not only with Boolean logic values but also time dependent values. The second class includes several component ageing models. Models included in these two classes are designed to be included in a RAVEN ensemble model to perform time dependent system reliability analysis (e.g., dynamic analysis). Similarly, these models can be interfaced with system analysis codes within RAVEN to determine failure time of systems and evaluate accident progression (static analysis).
- **TEAL:** This package enables the capability to compute the Net Present Value (NPV), Internal Rate of Return (IRR), and Profitability Index (PI) with RAVEN. Furthermore, it is possible to do an NPV, IRR, or PI search. TEAL will compute a multiplicative value (e.g., the production cost) so that the NPV, IRR, or PI has a desired value. The plugin allows for a generic definition

of cash flows with drivers provided by RAVEN. Furthermore, TEAL includes flexible options to deal with taxes, inflation, and discounting, and offers capabilities to compute a combined cash flow for components with different component lives.

- **VERT:** The Versatile Economic Risk Tool (VERT) is a model library to perform GRA. VERT quickly and effectively evaluates the economic risk that systems and sub-systems contribute to NPP power production. This is performed by employing classical PRA tools, such as FTs, with component reliability/availability models to evaluate risk associated with production loss.
- **SRAW:** The System Risk Analysis Workflows (SRAW) plugin for the RAVEN code is a plugin that enables RAVEN to perform stochastic analysis of NPP asset management. The primary function of SRAW is to generate the complex RAVEN workflows as necessary to optimize asset management under various scenarios.

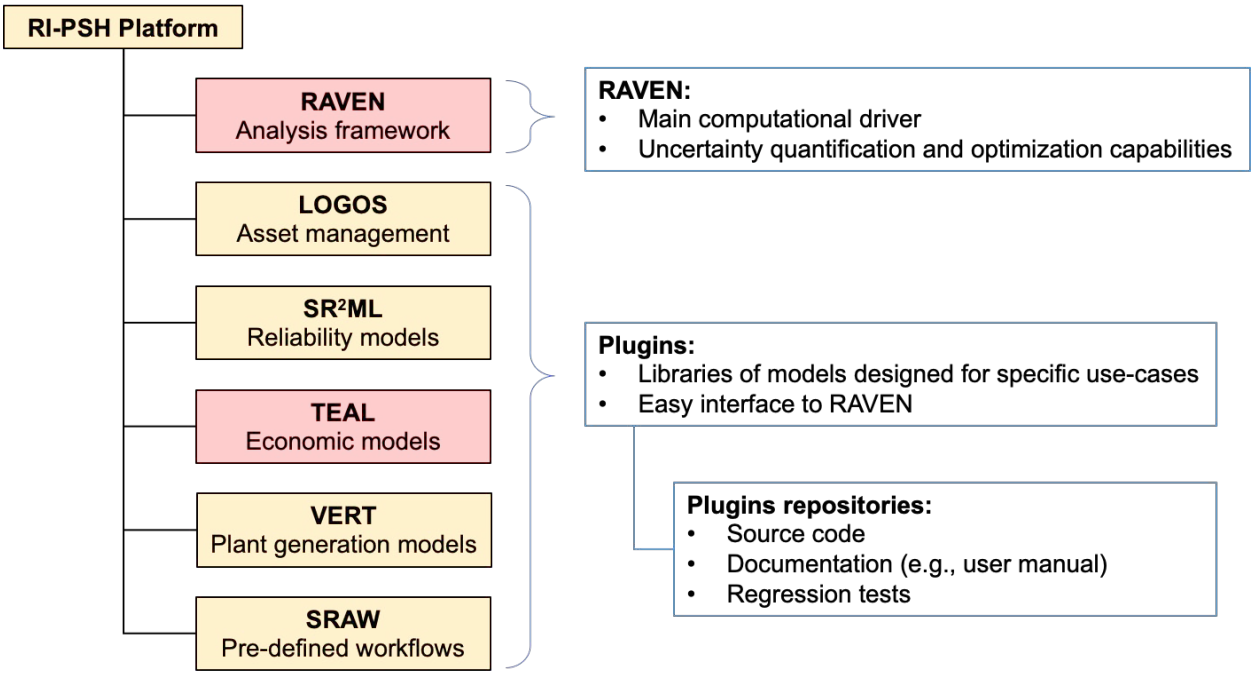


Figure 4. Structure of the RI-PSH computational platform.

While some repositories are already available using an open-source license (e.g., RAVEN and TEAL), the vision is to release all of the plugins with an open-source license sometimes in FY-21. This will greatly improve collaboration and development with external collaborators; this approach will support rapid deployment across the industry as well as provide a platform that permits expansion and improvements based on stakeholder experiences with use.

3. MAINTENANCE APPROACHES

The primary objective of a plant maintenance program is to effectively and efficiently maintain plant SSCs so that plant safety and production are maximized in a manner that is cost effective. Operating NPPs have successfully achieved high levels of safety (as indicated by numerous measures such as data and statistics maintained by regulatory authorities (e.g., United States Nuclear Regulatory Commission – US

NRC) and industry oversight organizations (e.g., Institute of Nuclear Power Operations – INPO) and production (as indicated by average plant capacity factors ~90% for more than the past 10 years).

However, achieving these levels of performance has come at a high cost in terms of maintenance expenditures. These high expenditures, combined with the implementation of required safety upgrades as a result of the Fukushima Daiichi accident in Japan, have challenged the economic viability of operating NPPs, with a number having been prematurely shutdown and decommissioned in the past several years due to an inability to compete economically. The objective of this research activity is to develop an integrated asset management approach to support plant decision-makers to optimize plant maintenance, testing, and surveillance activities to reduce costs and improve plant economics while simultaneously maintaining plant safety and production at the highest levels achievable.

3.1 Maintenance Activities

Consider maintenance alternatives at the component level. Then maintenance can be classified into one of four categories which are described below. The categories are intended to ensure planned maintenance is optimally cost-effective given the functional significance of a particular SSCs and available technological capabilities.

3.1.1 Corrective Maintenance (run to failure)

Corrective Maintenance (CM) activities applies to plant SSCs that do not meet the definition of "Critical Components" as specified in the "Critical Component Reduction" section of NEI EB-16-25 [2], as indicated in the "Equipment Reliability Process Description" section of INPO AP-913 [3]. These SSCs are typically designated as "run to maintenance" (RTM) in plant equipment databases. Note that the term "RTM" was previously referred to as "run to failure" (RTF). Both terms are considered synonymous in this discussion. For these SSCs, handling can depend on the approach a utility takes to address RTM SSCs, as observed in Figure 1.

As described in Section 3.1.5 of AP-913 Rev 6 [3], if RTM SSCs are included within the plant's population of non-critical SSCs, then some maintenance of that equipment may be included within the formal PM program with some failure mechanisms addressed by Preventive Maintenance (PM) strategies. However, if RTM SSCs are excluded from the set of SSCs classified as noncritical (i.e., RTM SSCs constitute a separate category), then these SSCs should not have any PM activities performed. Note that in this case there may still be simple cost-effective tasks used to extend the life of the SSC, however, they are either not included in a formal PM plant program, or the SSC can be run until CM is required.

From the perspective of asset management, the fundamental issue related to RTM SSCs is in the cost benefit tradeoffs related to the decision to allow a particular SSC to only be repaired when it fails. Since there are minimal impacts to safety or production upon failure of RTM SSCs, the economic impact related to these SSCs is minimal. As a result, these SSCs will not be addressed in long-term plant asset management plans.

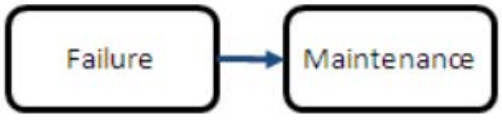


Figure 5. CM scheme [1].

3.1.2 Preventive Maintenance (Optimal Maintenance Interval)

In this category of maintenance alternatives, as well as the subsequent categories described in this report, the choices of which maintenance alternatives will be adopted are influenced by a combination of factors as seen in Figure 6. These include (1) whether a particular SSC is classified as a critical or a non-critical SSC, (2) the characteristics of the particular SSCs (e.g., what possible maintenance activities can be performed on the SSC to address the different identified failure mechanisms within the context of plant capabilities to implement the activity); and (3) economic considerations. It is generally recognized that more comprehensive maintenance activities will be specified for SSCs designated as critical than for those that are designated as noncritical. Finally, these decisions are also impacted by constraints that may be imposed (e.g., whether the activity requires a plant outage or whether it can be performed during power operation).

The PM activities in this category generally can be viewed from the perspective of periodic maintenance activities which historically have been performed at defined intervals of time. Examples include activities such as replacing consumable materials (e.g., performing an oil change) at a specified period of time or based on accumulated run time for the SSC and performing calibrations of various instrumentation that measures and controls plant process parameters. Many required activities that are specified in plant Technical Specifications that are related to operational testing of plant SSCs critical to plant safety are of this class.

These activities constitute those that are the most intrusive which are performed on a recurring basis for plant SSCs. They also typically possess greater costs than those in the Condition-Based Maintenance (CBM) and Predictive Maintenance (PdM) categories described below. Because these activities are intrusive, they also have the potential to leave the SSC in a worse condition (e.g., due to human error, introduction of defective replacement parts, etc.) than prior to the conduct of the maintenance. As a result, this class of maintenance alternative is less desirable than use of CBM or PdM when these approaches are available. Elimination of this type of activity (i.e., High-Cost Noncritical PMs) was a significant objective of the transition to Value-Based Maintenance (VBM) described in NEI EB-17-03a [4].

For the use of time-based PM, the fundamental issue related to this class of activities (assuming that neither CBM nor PdM are possible as cost-effective alternatives) is the determination of an optimal frequency of performance that balances SSC performance objectives and costs. In EB-17-03a, this was specifically identified by specifying the need to “integrate cost-benefit analysis into station work order and PM change processes (and) ensure that the PM change processes provide guidance to support development of maintenance strategies that maintain the required ER for least cost”. For the long-term management of plant assets, there are numerous occurrences of this type of maintenance activity to ensure the long-term safe and economic operation of the plant. Often this type of activity is related to evaluating the condition of plant structural members and ensuring that they remain in good condition to ensure the long-term safety and economic performance of the station. An example related to safety is ensuring acceptable structural integrity and condition of the spent fuel pool while an example related to production and economics is to ensure the proper functioning of condenser tubes trough the remaining life of the plant.

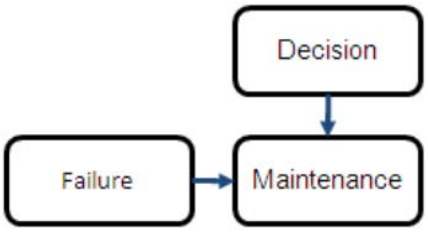


Figure 6. PM scheme [1].

3.1.3 Condition Based Maintenance (Observed Conditions)

The primary determinants related to the CBM category are: (1) the extent to which one or more technologies are available to assess the operational condition and performance of a particular SSC in relation to its expected failure mechanisms and (2) the extent to which application of these technologies can be cost-justified.

This category includes various types of monitoring and evaluation techniques as observed in Figure 7. As an example, for rotating equipment CBM often includes: (1) performance testing; (2) vibration analysis; (3) ferrographic oil analysis; and (4) infrared (IR) thermographic analysis. In this category, the CBM data provide a useful "snapshot" of the condition of the monitored SSC at the time the data were obtained. In this category, evaluation of the data generally consists of a comparison of the various data values to predefined "Alert" or "Action" thresholds, trending of data over time, and an evaluation against previously identified degradation indicators via the application of pattern recognition algorithms. In this category, the activities are diagnostic in nature (i.e., they generally are limited to being able to detect degradations in SSC condition or performance, albeit often at an early stage of degradation). However, this maintenance category is limited to the extent to which the applied monitoring can predict the course of future degradation over time.

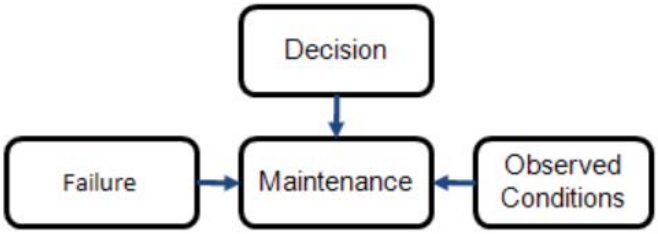


Figure 7. CBM scheme [1].

3.1.4 Predictive Maintenance (Prognostic - RUL estimate)

Similar to the previous category, the primary determinants related to the PdM category are (1) the extent to which one or more technologies are available to assess the operational condition and performance of a particular SSC in relation to its expected failure mechanisms and (2) the extent to which application of these technologies can be cost justified as seen in Figure 8. This category often uses many of the same technologies applied in the CBM category.

The primary distinction between the PdM and CBM categories is the extent to which the technologies can provide a reliable estimate on the course of progression of any identified performance deficiencies with the ultimate objective of providing an estimate of the time required before an action must be taken to prevent the monitored SSC to reach a state where its performance becomes unacceptable (or where it fails completely). This domain of monitoring typically is described as "prognostics" with the estimated time needed for the performance of required maintenance intervention designated the Remaining Useful Life (RUL) of the monitored SSC.

This mode of monitoring is the most challenging because, to be successful, it typically requires (1) a detailed understanding of the mechanisms by which performance degradations can progress; (2) validated models for progression to failure; and (3) substantial amounts of monitoring data (often with streamed data at relatively high sample rates to be able to identify small additional changes in SSC performance). For these reasons, use of prognostics often are more expensive than the other categories of maintenance activities described previously, and therefore, typically are reserved for use on functionally critical SSCs or those that would have a high cost associated with failure. However, use of prognostics typically will provide the greatest amount of knowledge related to SSC condition and performance, provide detection of degradation at a very early stage, and thus, be capable of providing substantial lead time to plan and execute maintenance. It should be noted that as the cost of data acquisition and required computational power decrease, the application of this class of maintenance has become (and is anticipated to continue to be) useable over a broader range of plant SSCs.

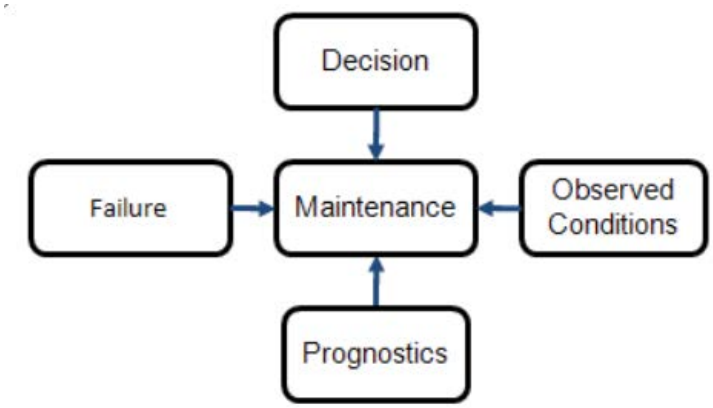


Figure 8. PdM scheme [1].

3.2 Classification of Maintenance Approaches

The approaches listed above can be classified in a 2-dimensional space as seen in Figure 9 based on the following characteristics:

- Underlying analysis method (data or model based)
- Type of deployment (on-line or off-line).

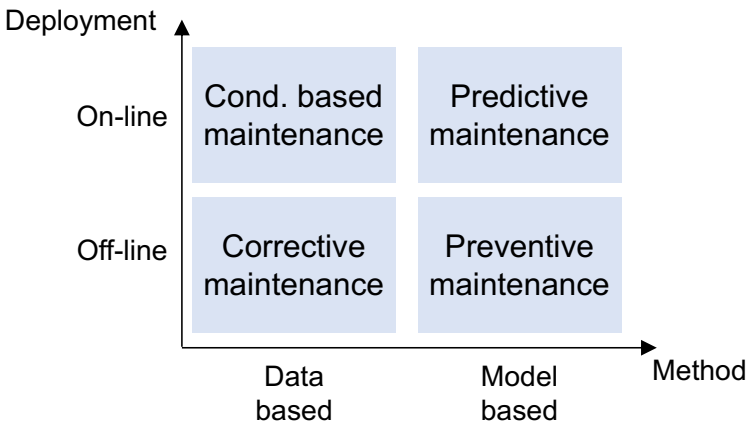


Figure 9. Classification scheme for the considered maintenance approaches.

In this classification, the analysis methods can be either qualitative or quantitative. In general, the off-line methods generally apply qualitative approaches to classify plant SSCs, especially for decisions related to the classification of SSCs designated as RTM. For both of these cases (e.g., CM for RTM SSCs and time-based PMs), any tasks and associated frequencies typically are specified via a combination of engineering judgement and plant / industry experience. Often, these activities are formulated into a standard PM program (i.e., PM templates).

Note that a significant variation of this approach is used for activities specified in plant TSs for which a plant may choose to apply a risk-informed approach to control the frequencies at which the activities are performed, such as with the use of a RI Surveillance Frequency Control Program (SFCP) [5]. A variety of methods have been applied within the industry to classify the level of functional importance of plant SSCs. A discussion of the history of the use of several of these approaches at commercial nuclear power plants for the purposes of improvement in plant ER programs is described in the report Plant Integral Risk-informed System Health Program (RI-PSH) [6].

Use of on-line methods generally requires quantitative evaluations that monitor, trend, and assess critical parameters of SSC condition and performance. As discussed previously, for the CBM approach, the analytical methods are generally straightforward, i.e. comparison of observed data to predefined "Alert" or "Action" thresholds, trending of data over time, and an evaluation using pattern recognition algorithms. Also, as previously indicated, the use of prognostics typically requires substantially more sophisticated analytical methods that address the relationship of the collected data to the monitored SSC degradation mechanisms, validated models that predict the SSC progression to failure, and larger amounts of monitoring data.

3.3 Matching SSCs to Appropriate Maintenance Approach

Given a specific plant system, which maintenance strategy should be assigned to each component contained in it? As indicated in Figure 10, the specification of maintenance activities can be viewed as a continuum where the least functionally important / costly SSCs are on the left while the most functionally important / expensive SSCs are on the right. Such a viewpoint is consistent with the process that is in use across the industry as discussed previously.

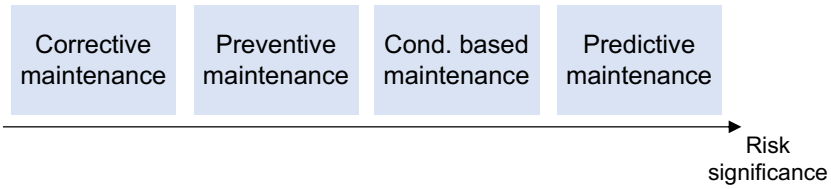


Figure 10. Component maintenance strategy and component risk significance.

Matching SSCs to the most cost-effective maintenance approach consists of a two-step process that identifies the equipment functional importance and then specifies the applicable maintenance tasks and frequencies based on a combination of this functional importance, SSC performance objectives, and costs. As described previously, the first element (e.g., specification of SSC functional importance) typically uses qualitative methods. However, quantitative methods can be employed to supplement these evaluations as appropriate. Example quantitative metrics that can be used to assess risks that are relevant to particular SSCs include the following:

- Fussell-Vesely (FV) importance, based on cut sets that relate to plant safety
- Prevention Worth (PW), based on path sets that relate to economic and generation risk

The use of the FV measure is a standard metric used by industry in SSC classification related to meeting maintenance rule requirements [7]. However, the use of PW has not been applied widely in the commercial nuclear industry. One reason for this has been that quantitative risk models that evaluate loss of production, such as GRA models, have not been developed as they have been considered to insufficiently provide value to justify the resource expenditures required to develop, evaluate, and maintain them.

4. MAINTENANCE MODELING

The objective of this section is to introduce current methods to probabilistically model component ageing and ER activities such as SSC maintenance, testing, and replacement/refurbishment.

4.1 Observations on Current Methods

The majority of maintenance modeling/optimization articles available in the literature present models and methods that rely on a large amount of available reliability information (i.e., data) in order to include several ER aspects (e.g., ageing, testing, maintenance) into the analysis. However, with few exceptions, the current fleet of operating NPPs possess data that are sparse with respect to utilizing many of these models. Additionally, when plants have attempted to collect such data in the past, this has proven to be a time-consuming, labor-intensive, and costly effort.

Some authors recognize such limitations and they attempt to solve it by including data uncertainties to compensate for the lack of the data itself as if data uncertainties were directly available. However, given the practical need of being able to support decision-making in a cost-effective manner, the question of what the point is in building complex (i.e., high fidelity) models/methods when the data required are too costly to obtain and that, even if they were available, they are affected by large uncertainties that would limit the potential capabilities and utility of such models/methods is a reasonable one to ask. As a result of this "vicious circle," the use of such models has not been widely adopted.

Another issue that is relevant to use of the models described in the literature include to what extent and, more particularly, how does an estimated (either single value or probability distribution) SSC probability

of failure within some time interval provide meaningful information to relevant decision makers? These questions are applicable at several levels. First, they are relevant for plant system engineers who are tasked with monitoring and managing the performance of their assigned SSCs. Second, they are also relevant to plant management who are responsible for evaluating, approving, and allocating plant maintenance and capital improvement budgets.

Given the overall objectives and constraints on operating NPPs and the known limitations in both traditional reliability modeling and available data, the fundamental question is how can one best utilize improvements in the state of knowledge and computational capabilities to develop a more cost-effective and integrated approach to system health and maintenance management? One solution to this problem would be to develop data-centric models/methods (i.e., models that are designed and built based on directly available data) that is designed to provide information that is useful to the various stakeholders (e.g., plant system engineers and management decision makers). It should be noted that until recently such a data centric approach was beyond the technical capability of existing computational systems. However, the increased capabilities and reduced costs of computational systems (both hardware and software – including use of advanced techniques such as artificial intelligence, data analytics, and machine learning) combined with advances in integrated data acquisition systems / platforms make such approaches much more feasible and cost effective[b].

This section describes an alternate approach for ER. However, before describing the approach, we first discuss the relevant issues:

- Which types of data are available from the various sources at the plant?
- What specific information is needed to support the various responsible stakeholders (i.e., plant system engineers and management decision makers)?
- Given these "boundary conditions" what methods and tools are available to most effectively and efficiently develop information to meet these needs?

4.2 ER Data Types

In the assessment of equipment and system performance, a large amount of data exists that have been generated over the course of a plant's operating lifetime. Because the fleet of operating NPPs in the U.S. was designed in the 1960s and primarily built during the 1970s, the vast majority of data on plant equipment performance was generated from analog instrumentation and either recorded on paper media (e.g., plant surveillance test results) or hard copy strip charts. Although some of the instrumentation associated with plant systems has been replaced with digital instrumentation and controls—in particular for the feedwater and main turbine control systems—and many plants have added data highways for remote monitoring and assessment of SSC performance and condition, often in a centrally located maintenance and diagnostics center, this situation still exists for a large portion of plant SSCs.

The type and amount of data available to assess SSC condition and performance (i.e., for the purposes of evaluating and managing system health) are numerous. Specific data sources also vary significantly across plants throughout the industry. However, there are several sources of data that are relatively standard. These data sources can be classified as one of two types: dynamic and static. Dynamic data sources represent those that capture "real time" data that are updated frequently; such data, when they exist, often are streamed to a centralized location for evaluation and analysis using various automated techniques, such as pattern recognition, statistical trending, etc., to identify changes in SSC performance at an incipient stage.

[b] As an example, from a software point of view, several data analysis (such as: numpy-scipy, pandas, seaborn) and machine learning libraries (such as: scikit-learn, TensorFlow, Keras, Theano, PyTorch) has been developed and released open-source.

These data can also then be used (often when combined with additional data of either type) to provide predictions of the evolution of SSC performance with the intended outcomes being to identify the specific cause of the degradation and if and when SSC failure would be anticipated to occur so that appropriate maintenance actions can be proactively scheduled.

The second type of data source consists of static data. These data typically are contained in plant databases (e.g., the most significant of which is the integrated plant work management system). These large data sources support a multitude of plant functions including work planning and management, corrective action programs, and regulatory compliance. For the purposes of system health and asset management programs, the most significant are the following:

- Maintenance reports (including SSC maintenance work orders documenting both corrective and preventive maintenance actions, maintenance rule SSC availability and reliability monitoring results, etc.)
- Surveillance reports (including SSC performance information and As Found / As Left data for instrument calibrations)
- Reactor Operator and Shift Supervisor operating logs
- Condition monitoring data (including data sent to a centralized maintenance and diagnostics center)
- Other monitoring and testing data (including quantitative and qualitative SSC condition data obtained from manual data gathering such as off-line vibration monitoring, thermographic imaging, lubricating oil analysis, etc.)
- Other reports (including engineering analyses, plant corrective action program reports, etc.)

Note that the format and accessibility of these data sources (and others) are different for different plants and over differing time frames. As one example, instrument calibration records (with As Found / As Left data) that are necessary for some engineering evaluations (such as surveillance frequency extensions performed under a plant SFCP) may be available only on hard copy (paper) version of the completed surveillance tests. In other instances, some data may only be available in electronic format after the date of installation of the applicable database in which is the data are contained.

4.3 ER Data Issues

Although a large amount of data generally is available to support plant ER and asset management programs and decisions, the data are often contained in different data sources (some of which may not be in electronic format) and may have other limitations that may limit their usefulness. Some of the issues that complicate the ability to use these data effectively are the following:

- *Data scarcity*: Specific data that are best suited to address specific issues may not be available for use for a variety of reasons. Often this is the case due to not having specific process conditions monitored to provide direct indication of the desired information.
- *Data sparsity*: A significant portion of plant SSCs that are deployed in operating NPPs are for use to respond to accident conditions. As a result, these SSCs are normally in a standby condition with data only obtained periodically when the SSCs are tested to ensure proper operation and meeting requirements incorporated in the plant TSs.
- *Incomplete data*: Data may be difficult or expensive to retrieve, particularly for data that were collected early in the plant operating history.

- *Contradictory data*: In addition to potential gaps in data, some data may provide information that appears to be contradictory and for which the original analyses that addressed such discrepancies is not available.
- *Heterogenous formats and contents*: This issue may be the most significant impediment to the cost effectiveness of plant ER and asset management evaluations and decision making. In addition to requiring significant resources to identify and obtain necessary data to support effective and efficient decision making, data are often contained in different formats and stored on different media. Since digital media were not available early in the life of many operating NPPs, early records often have been maintained in hard copy (paper) versions or stored on microfiche. Additionally, as digital media became available and matured, data stored in digital format often migrated across several platforms as technology and related applications changed over the years.
- *Data fuzziness and uncertainties*: In some instances, uncertainties in collected data are not recorded or addressed. In general, this usually is not a significant limitation; however, it may become an issue in addressing some regulatory applications (e.g., in applications that are risk-informed).

The extent to which these (and potentially other) data issues may be significant depending upon both the application (i.e., the specific decision being addressed) and the specific structures, amount, and content of the various plant data sources. The importance of these issues may be increased substantially if they are intended to apply artificial intelligence and machine learning approaches to support the analysis of available data and provide input into the plant decision-making process.

4.4 System Engineer Tasks, Decision-Making and Required Knowledge

The main tasks of a plant system engineer can be summarized as follows:

- Monitor and gather information on the status and performance of assigned plant SSCs
- Plan, execute, and manage SSC ER activities for assigned plant SSCs
- Develop ageing management plans for assigned plant SSCs
- Develop and manage budgets for assigned plant SSCs

The models/methods under development for the RIAM-PHM projects are intended to directly support cost effective performance of these tasks. Whiel determining an approach to achieve this objective, two questions, based on the accumulated experience of NPP operation are asked:

1. *Does a probability of-failure-based language provide a useful to support these decisions?*

   Typically, the answer to this question can be considered to be no. Although portions of a reliability engineering based approach have been implemented to meet industry guidance related to the maintenance rule [7], most of these formal methods have been found to be costly, time consuming, and labor intensive to apply. As one example, in the late 1980's / early 1990's formal Reliability Centered Maintenance (RCM) approaches were adapted from use in commercial aviation and applied to NPP maintenance programs. However, application to industry pilot plants found the approach to be very labor intensive and costly. This led to development and implementation of streamlined RCM (so called SRCM) approaches and software, culminating in development of the EPRI PM Basis Database software package with

standard maintenance programs (so called PM templates) to facilitate more cost effective specification and management of plant PM programs [8, 9]. Another example is the conduct of system/train level evaluations of reliability. Such evaluations are performed to meet industry guidance to comply with the maintenance rule and for evaluation of plant PRA models. However, such evaluations have been limited to use in these applications as they also have been found to be labor intensive and costly to perform. Industry efforts to apply these techniques to evaluate the potential for plant trips (e.g., development of plant GRA models were abandoned as being too costly and not providing sufficient value for the given expenditures that would be needed to build, validate, and maintain them.

2. *Given industry experience, what would be a suitable framework that would be useful for a plant system engineer?*

   In the operation of commercial NPPs, the concept of margin management has served as a useful tool to maintain safe operation. Similar to other engineering applications where it is considered necessary to ensure that adequate levels of safety are provided, the concept of safety margins has been employed within the NPP regulatory framework since the beginning of the industry. Because this approach is well understood and permeates the decision processes used within the industry, a useful approach to manage plant ER and system health that is based on evaluation of "margin to failure" would be effective and efficient. Such an approach is proposed for use by plant system engineers and is described below for each of the maintenance classes.

5. MARGIN TO FAILURE BASED APPROACH

This section presents an alternate "language" to perform reliability modeling as an alternate path to failure-based reliability modeling. This new language is based on a concept that is more familiar to plant system engineers and management decision makers, i.e., the concept of margin to failure.

5.1 A System Engineer Definition of Risk

In its classical definition, risk is defined by three elements: what can go wrong, what are its consequences and how likely is it to occur? Likelihood of occurrence of events is typically described in probabilistic terms. While this choice makes sense in a regulatory based framework to estimate risk associated to NPPs in terms of Core Damage Frequency (CDF) and Large Early Release Frequency (LERF) as evaluated in a plant PRA, this approach, which as implemented thorough the industry relies on static modes as thus does not provide an actual snapshot of the health of plant.

Several observations are provided to support this claim:

- Plant PRA models are based on Boolean logic structures (e.g., Event Trees [ETs] and FTs) which describe the deterministic functional relationship between systems, components and human interventions
- Each basic event in a PRA represent a specific elemental occurrence of an event (e.g., failure of a component, failure to perform an action by the plant operators, recovery of a safety system)
- A basic event is defined using a Boolean logic, i.e., the event can either occur or not occur
- A probability value is associated to each basic event, which represents the probability that the basic event can occur

- Testing, maintenance and surveillance operations are not completely integrated into a PRA structure
- This probability value is updated typically every two years based on past operational experience through use of a Bayesian statistical process
- A probability value associated to an event is thus an integral representation of the past operational experience for such an event and it neglects the present health status of components (from diagnostic and CBM data) and health projection (when available from prognostic data) on anticipated changes in SSC condition and performance in the near future[c]

Given these conditions, the decision-making process related to health and asset management is not well suited to being evaluated and managed using classical PRA tools. What can we change then? A possible alternate path can start by redefining the word "risk" to a broader meaning that better reflects the needs of a system health and asset management decision making process. Figure 11 below shows this alternate path where, rather than asking how likely an event can occur (in probabilistic terms), we ask how far this event is from occurring.

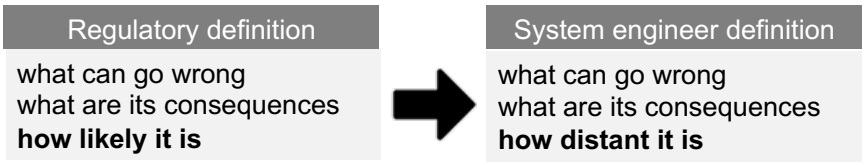


Figure 11. Regulatory and system engineer definition of risk.

This new interpretation of risk transforms the concept from one that focuses on probability of occurrence to one that focuses on assessing how far away (or close to) an SSC is to an unacceptable level of performance or failure. This transformation has the advantage that it provides a direct link between the SSC health evaluation process and standard plant processes used to manage plant performance (e.g., the plant maintenance and budgeting processes). The transformation also places the question into a form that is more familiar and readily understandable to plant system engineers and decision makers.

At this point it is needed is to identify how to measure SSC margins as a distance of occurrence between the SSCs existing condition and the point where its performance or condition becomes unacceptable and action is required. Note that the concept of "distance" does not necessarily need to be measured in terms of time as we will clearly show in the next sections. Now, rather than using a "failure" language, we will measure this distance in terms of margins: margin to failure. Given the data available from plant ER and monitoring/diagnostic-prognostic centers, a margin to failure can be described in either probabilistic or single-value forms.

5.2 Margin to Failure

As a first-tier definition a margin can be viewed as follows:

Definition: Margin to failure (MTF) - The "distance" between present/actual status and a failed status for a specific plant SSC (see Figure 12).

[c] In this respect, when a failure mode of a component is subject to monitoring (e.g., through diagnostic/prognostic process) and maintenance operations, the aleatory concept of failure rate might be inadequate to model such failure mode.

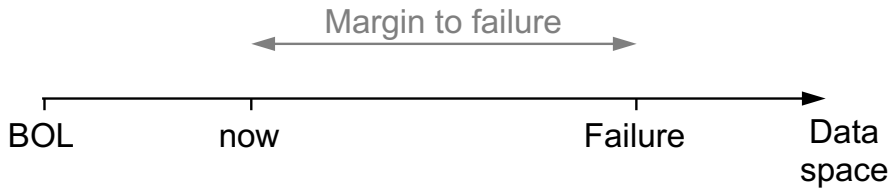


Figure 12. Graphical representation of margin to failure.

The application of this framework to plant ER and system health as proposed here is centered on the integration and evaluation of data that are available to assess SSC condition and performance. Thus, this framework requires the definition of the following concepts:

- *Space*: The "space" definition should be based solely on the type/dimensionality of the data that can be directly measured/obtained by the system engineer
- *Distance metric*: Once the space is defined it is necessary to provide a measure of the "distance" between two points located in this space.

The goal now is to define appropriate "Space" and "Distance" metrics for the four classes of maintenance approaches identified previously: CM, PM, CBM, and PdM (see Sections 5.2.1 through 5.2.4). In each case the objective is to develop measures that provide useful indicators of SSC condition and performance which can be used by plant system engineers to cost-effectively assess and manage their assigned SSCs.

The writeup provide in the following sections for each maintenance class identifies available data that are relevant to the class and the space in which the relevant data are applicable. Then for each maintenance class the relevant measure of MTF is identified. In each case the approach is illustrated via a graphical depiction.

5.2.1 Corrective Maintenance

This class represents the simplest maintenance class for which the SSCs assigned to the class have been identified as not critical to plant production or safety. Since the SSCs in this class are allowed to run to maintenance, at which time they are repaired or replaced as necessary, the only information available is the time at which a degraded condition was detected or a failure occurred that resulted in generation of the CM work order to restore the functionality / condition of the SSC.

For the purposes of estimating the MTF for a particular SSC, this can be obtained from the difference in time between when an SSC was placed in service and the Mean Time To Failure (MTTF) for a population of similar SSCs operated under similar conditions as observed in Figure 13. Because the SSCs in this maintenance class are not functionally critical to plant operation or safety, precise estimates of these times (and of the estimated MTF) are not critical for normal operational purposes. However, for purposes of long-term asset management, such estimates would be useful to identify appropriate maintenance intervals (such as in the case of replacement to address issues of obsolescence) for particular classes of SSCs (e.g., replacement of various process sensors and electronic signal converters) as investments to support cost-effective operation during periods of license extension:

Data: Observed failure times for similar components situated within similar boundary conditions

Space: Time

MTF: Distance between actual operation time since SSC installation and the set of observed failure times

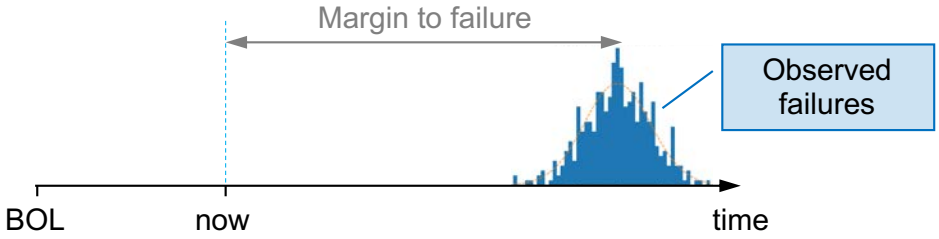


Figure 13. MTF in a CM context.

5.2.2 Preventive Maintenance

As described previously, this maintenance class predominantly consists of those time directed PM activities that are intended to address identified degradation mechanisms before they can result in failure of the SSC. Since these activities are scheduled at specific instances of time, for the purposes of estimating MTF we indicate ΔT_{PM} as the time between two consecutive PM activities (i.e., the PM frequency) that address the same or similar failure mechanisms.

Similar to the previous case, the relevant data are the times when a degraded condition was detected or a failure occurred that resulted in generation of the CM WO to restore the functionality/condition of the SSC; however, because these SSCs have PM activities that are intended to preclude failure, such occurrences should be infrequent. Because such failures for these SSCs should be infrequent, an additional source of data comes from the conditions observed when PM activities are performed.

From these data, three outcomes of significance are observed: (i) significant deterioration in SSC performance or condition, which is indicative of the need to perform more frequent or different PM; (ii) minor but noticeable observed deterioration in SSC performance or condition, which is indicative that the prescribed PM tasks and frequencies are appropriate to cost-effectively manage SSC health; and (iii) minimal to no observed deterioration in SSC performance or condition, which is indicative that the specified PM activities do not need to be performed as frequently as currently specified, as shown in Figure 14:

Data: Failure times observed within a PM interval (i.e., between PM_i and PM_{i+1})

Space: Time

MTF: If failure before PM_{i+1} has been observed, then MTF is the distance between present time and the recorded failures. If no failures have been recorded, then one can set $MTF = \Delta T_{PM}$. (Note that for data characterized by (iii) above, this will result in a conservative assessment as the presence of little to no degradation is evidence that the MTF is likely substantially larger than ΔT_{PM}.) By combining these two cases one obtains $MTF = \min\left[distance(now, failure\ dist < PM_{i+1}), \Delta T_{PM}\right]$]

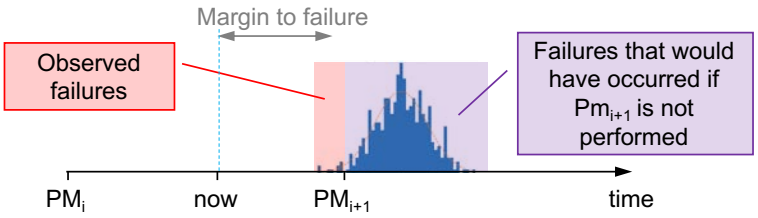


Figure 14. MTF in a PM context.

5.2.3 Condition-Based Maintenance: Diagnostic

For this (and the following) maintenance class, data are available that evaluate the condition and performance of the monitored SSCs. Often these data are available in real time leading to the capability of performing real time assessments of SSC condition and performance. As described previously, in this category the data evaluation approaches are diagnostic in nature but are limited in the extent to which the applied monitoring can predict the course of future degradation over time.

Although there is a much better understanding of the condition and performance of monitored SSCs in this maintenance class, there remains substantial uncertainty in the progression of the active failure mechanisms that are occurring. Therefore, once degraded performance is detected on a particular SSC, it is prudent to perform appropriate actions as soon as practicable to reduce the likelihood that the degradation mechanisms progress faster than anticipated leading to SSC failure since SSCs within this class are critical (from either an economic or safety perspective). Note that in this class, the identified actions will likely be conservative given the importance of the SSCs and the uncertainties in the progression of the relevant degradation mechanisms as observed in Figure 15:

Data: Actual SSC condition and past condition data for similar SSCs

Space: SSC condition

MTF: Distance between actual SSC condition and observed SSC past conditions that lead to failure

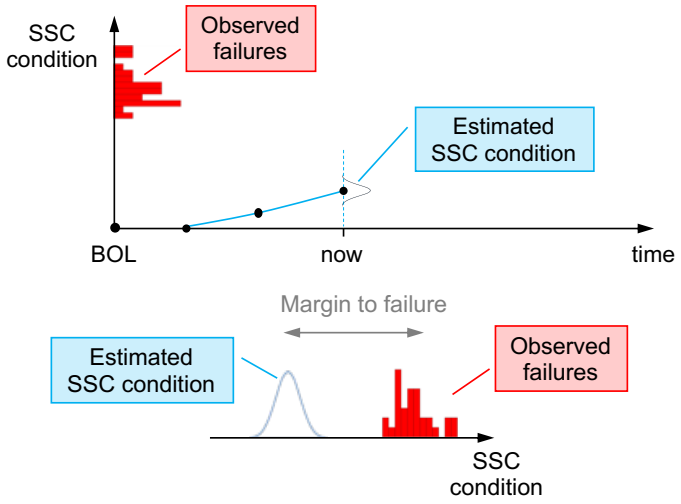


Figure 15. MTF in a CBM context: evolution as a function of time (top) and a graphical plot of MTF (bottom).

5.2.4 Predictive Maintenance: Diagnostic + Prognostic

Similar to the preceding class, data for SSCs in this maintenance class are available to evaluate their condition and performance, again often in real time. However, as described previously, in this category the approaches are prognostic in nature and can provide an accurate prediction of the course of future degradation over time.

Therefore, once degraded performance is detected on a particular SSC, appropriate action is to apply the relevant models using the given data to obtain predictions of the MTF for the SSCs and integrate these evaluations (including any identified contingency actions such as the conduct of additional or more sophisticated / targeted monitoring).

Note that in this class, due to the higher levels of confidence (e.g., smaller levels of uncertainties) in predicting future SSC degradation, the identified actions may not need to be as aggressive or conservative as those implemented for SSCs in the previous maintenance category as seen in Figure 16:

Data: Actual SSC condition and past condition data for similar SSCs and predictive model of future degradation (which provides the estimated RUL for the monitored SSC)

Space: Time

MTF: Distance between actual time and estimated RUL

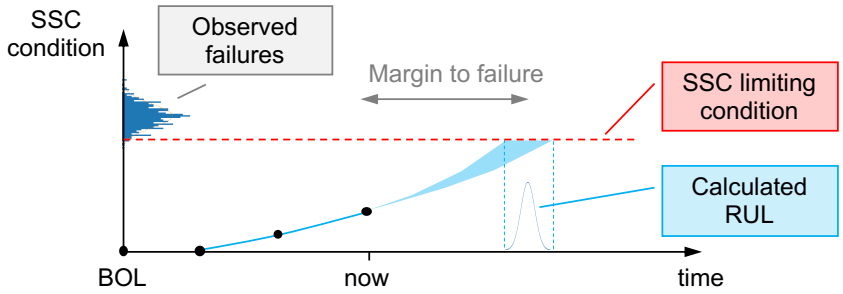


Figure 16. MTF in a PdM context.

5.3 Normalized Margin to Failure

The definitions of MTF indicated in Section 5.2 are defined over the different spaces that are reflective of the different maintenance classes.

The goal now is to answer this question: is there a way to transform the definitions of MTF indicated in Sections 5.2.1 through 5.2.4 in such a way that they can be compared? A possible approach is to normalize the MTF definitions in Sections 5.2.1 through 5.2.4 (i.e., use as a measure a normalized margin to failure [NMTF] where each measure is normalized over its specific attributes as indicated below).

- CM:

$$NMTF = \frac{MTF}{observed\ failure\ times} \quad (1)$$

- PM:

$$NMTF = \frac{MTF}{\Delta T_{PM}} \tag{2}$$

- CBM:

$$NMTF = \frac{MTF}{observed\ SSC\ conditions\ that\ lead\ to\ failure} \tag{3}$$

- PdM:

$$NMTF = \frac{MTF}{RUL} \tag{4}$$

This normalization results in a standardized evaluation set such that $0 \leq NMTF \leq 1$. In this framework, the NMTF provides relative assessment criteria that can be compared across the different maintenance classes and for which consistent evaluation and decision criteria can be developed.

5.4 Reliability Models for NMTF

Currently, applications of a number of reliability models are under development and evaluation (e.g., GRA models) as a part of the LWRS program. These models are being constructed using standard reliability approaches that have been used in commercial nuclear power applications (i.e., ET/FT models) for many years and, as such, are based on the assumptions that each basic event is defined over a probability of failure value.

The goal now is to deploy these reliability models by feeding NMTF values at the component level and propagate these results through the models in order to determine NMTF at a level where the degraded performance or failures would result in actual consequences to plant safety or economics (i.e., the system or train level).

Even though the definition of NMTF is between 0 and 1, note that it is not appropriate to simply exchange probability values with NMTF values; in other words, it does not make sense to add or multiply NMTF values like it has done with probability values.

5.4.1 Mathematical Basis for NMTF Based Modelling

Currently plant reliability models (including PRA and GRA models) are based on Boolean logic structures (e.g., ET and FT) and are solved probabilistically using classical probabilistic calculations applied to the sets as observed in Figure 17.

As shown in Figure 17, provided a set A in a sample space S, we can associate a probability value $P(A)$ to such a set such that $0 \leq P(A) \leq 1$, where $P(\emptyset) = 0$ (here $\emptyset$ indicates the empty set) and $P(S) = 1$. From a set theoretic perspective, it is of interest to measure (in a probabilistic sense) the probability associated with the occurrence of both (intersection of A and B) or either (union of A and B) events. While the union of events maps the region in S covered by both A and B, their intersection maps the overlap region between A and B.

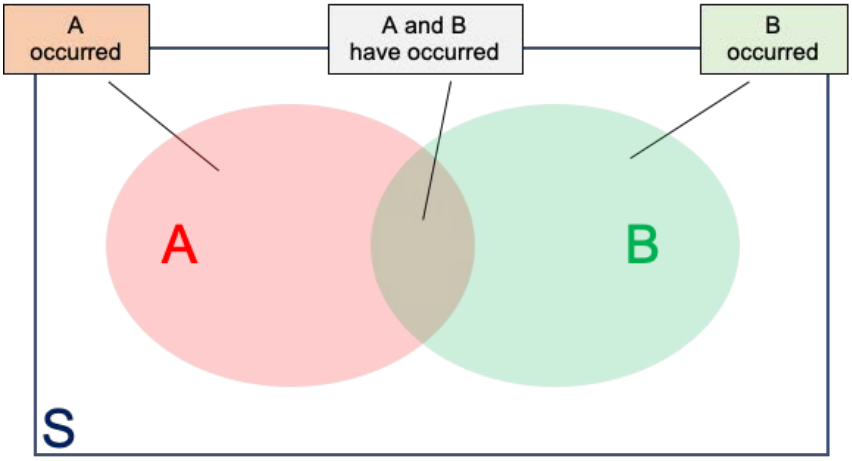


Figure 17. Classical set representation of event occurrences.

Given this, it is possible to calculate the probability of the union and intersection of two events A and B as follows:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$
$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A) \quad (5)$$

where the notation $P(A|B)$ indicates the conditional probability of the occurrence of A given that B has occurred.

Provided the assumptions that A and B are independent, then these equations reduce to the following:

$$P(A \cup B) = P(A) + P(B) - P(A) \cdot P(B)$$
$$P(A \cap B) = P(A) \cdot P(B) \quad (6)$$

A similar discussion can be carried out in terms of NMTF. Consider two components (A and B). The NMTF for both components can be visualized in a 2-dimensional space as shown in Figure 18.

Starting with brand-new components (i.e., $NMTF_A = NMTF_B = 1$), aging degradation that affect both can be represented by the blue line of Figure 18 which moves from the point of the coordinates (1,1), components A and B at the beginning of life (BOL) to the point of the coordinates (0,0), both components failed. Each point in the blue line represents the $NMTF$ for both components A and B at a particular time instant.

Similar to the set-based visualization of Figure 17, we can identify the following regions in Figure 18:

- Occurrence of both events: when $NMTF_A = 0$ and $NMTF_B = 0$
- Occurrence of either events: when $NMTF_A = 0$ or $NMTF_B = 0$

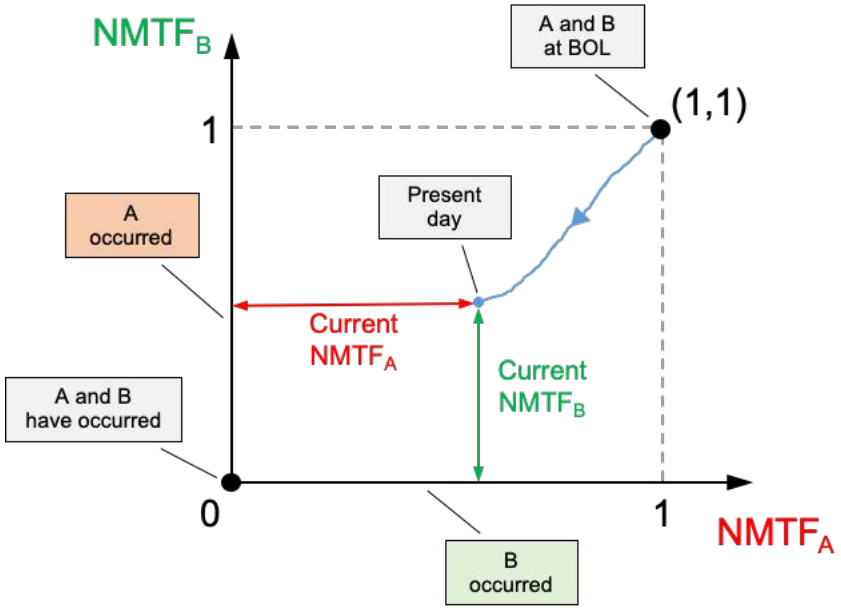


Figure 18. Graphical representation of event occurrences based on NMTF.

At this point we can calculate the NMTF for the events listed above. This is accomplished by following the definition of MTF: by measuring the distance between the actual condition of components A and B and the NMTF conditions identified by the event under consideration (e.g., the occurrence of both or either events):

- NMTF(occurrence of both events): $dist[(NMTF_A^{act}, NMTF_B^{act}), (0,0)]$
- NMTF(occurrence of either events): $min(NMTF_A^{act}, NMTF_B^{act})$

The function $dist[X, Y]$ is designed to calculate the distance between two points X and Y. Like for any other n-dimensional continuous space, several distance metrics can be chosen as indicated in Table 1.

Table 1. Summary of possible distance metrics.

Name	Formula
Manhattan	$dist[X,Y] = \sum_{i=1}^{n} \lvert X_i - Y_i \rvert$
Euclidean	$dist[X,Y] = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2}$
Chebyshev	$dist[X,Y] = \max_i \lvert X_i - Y_i \rvert$

5.4.2 Examples of NMTF Temporal Evolution

Provided ER data that monitor the health and condition of a plant SSC, the NMTF of the component evolves in time. Note that the occurrence of degradation can involve multiple competing failure modes for a single component. Figure 19 shows a hypothetical temporal evolution for two components or, alternatively, two failure modes for a single component. Note that the illustrations used here are simplified to a two-dimensional representation to illustrate the concept. In actual application to plant SSCs, the evaluations would occur over a (potentially much) larger dimensionality.

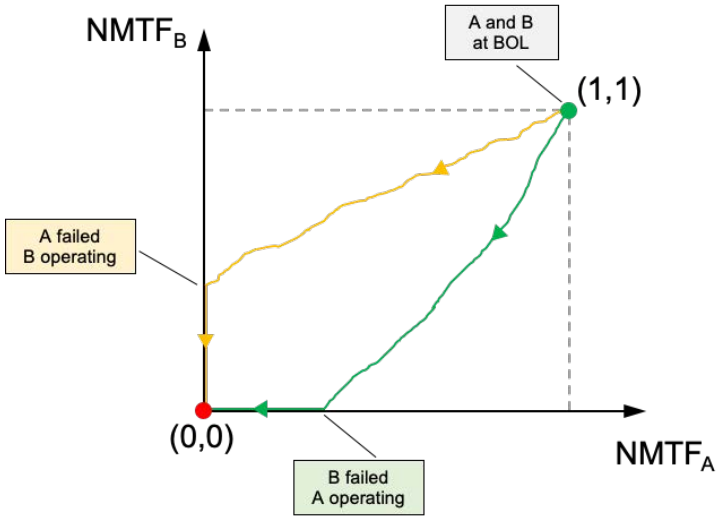


Figure 19. Temporal evolution of NMTF for two components.

Maintenance activities are designed to improve component health and, thus, renew its NMTF (by resetting NMTF to the point (1,1) in this example). Table 2 provides an example of activities that can be performed on an electric motor to check, test, and improve its reliability.

Table 2. Some electric motor surveillance, testing and maintenance activities.

Surveillance	Testing	Maintenance
Vibration monitoring	rpm vs. torque curve update	Seal replacement
Temperature monitoring	Motor winding insulation resistance test	Change oil in seal and motor housing
Check oil levels in seal and motor housing	Motor winding continuity test	Bearings replacement

These activities can be incorporated in a NMTF reliability analysis by incorporating testing and surveillance data to determine the status of component (or individual failure modes) NMTF. Maintenance activities (e.g., PM) are designed to improve the component (or failure mode) NMTF and, hence, its

reliability. Figure 20 shows the impact of maintenance activities on the NTMF temporal evolution of Figure 19.

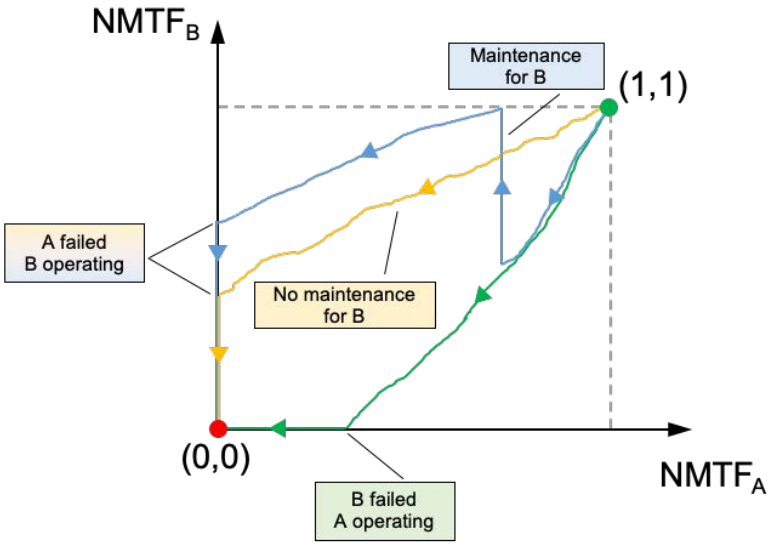


Figure 20. Impact of maintenance activities on the NMTF progression.

5.4.3 NMTF Based Simple Reliability Calculations

Note that the plant / system / train reliability models simply depict the logical connections between components and how the failure of a component affects system operation. Although the output of the model is probabilistic, the underlying structure is deterministic.

The question now is: is it possible maintain the same structure of the underlying reliability models but redefine / reinterpret the basic events in terms of NMTF? In other words, how can we calculate system / train / plant level NMTF values given specific component level NMTFs and the developed reliability models?

5.4.3.1 Series and Parallel Configurations

Let's consider a simple case consisting of two SSCs (SSC1 and SSC2) for two possible configurations: series and parallel. In the reliability model, two basic events are associated with each SSC, i.e., BE1 for SSC1 and BE2 for SSC2. Figure 21 depicts a parallel configuration of SSCs while Figure 22 depicts a series configuration. In these figures, TE represents the top event, i.e., the event being modeled which in these cases is the capability of the overall system to function. Similarly, BE represents a basic event, which in these two simple examples indicate whether or not SSC1 and SSC2 operate successfully.

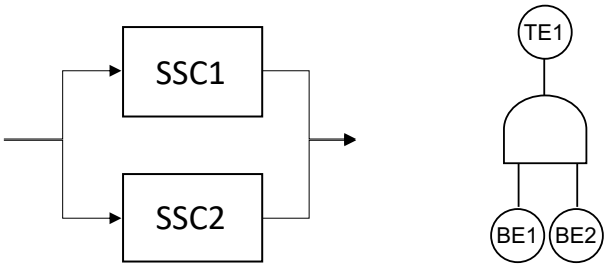


Figure 21. Parallel configuration and relative Boolean logic representation.

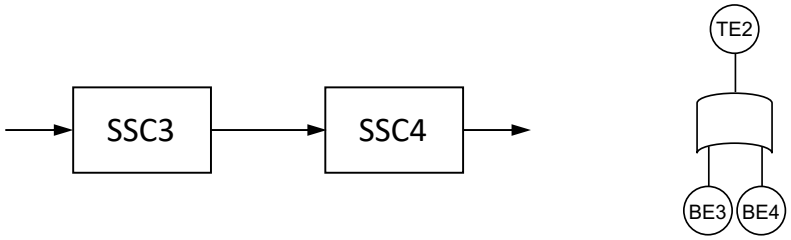


Figure 22. Series configuration and relative Boolean logic representation.

Instead of probability values for BE1 and BE2, NMTF1 and NMTF2 are provided. What is the NMTF for the system level top events TE1 and TE2 for this example? Given that NMTF provides information on how far a component is from failure, the NMTF for TE1 and TE2 can be calculated as follows for the two examples:

- Parallel configuration: System fails when both SSCs have failed, thus NMTF for TE1 is equal to the distance from the point $(NMTF_{BE1}^{act}, NMTF_{BE2}^{act})$ to the point $(0,0)$:

$$NMTF(TE1) = distance[NMTF_{BE1}^{act}, NMTF_{BE2}^{act}] \quad (7)$$

- Series configuration: System fails when at least one SSC has failed, thus NMTF for TE2 is driven by the component in the worse condition

$$NMTF(TE2) = min(NMTF_{BE3}^{act}, NMTF_{BE4}^{act}) \quad (8)$$

Therefore, it is seen that for this very simplified example, replacing failure probabilities of SSCs in a standard ET/FT model with NMTF results in the evaluation being simplified from the performance of Boolean operations on probability values to a much simpler process of evaluation of minimum and maximum values. This has the potential value of transforming the underlying reliability models from providing static results (e.g., the failure probabilities are generally evaluated as constants) to one where they can respond in a dynamic manner to the actual performance and condition of the underlying SSCs included within the models. Note that although these examples are extremely simple, as in the case of standard PRA models, they consist of the basic "building blocks" from which much more complete and comprehensive modes can be constructed. Additionally, the approach can make substantial use of existing plant modes (such as a PRA or GRA model). Such a framework has the potential to standardize and simplify plant decision making.

5.4.3.2 KooN Configuration

In this system configuration N components out of K are required to guarantee system function (see Figure 23). Such a configuration is commonly referred to as a K out of N (or KooN) configuration. Let's assume the first K components ($BE_1, \dots, BE_K$) have the highest values of NMTF. By following the reasoning shown in the previous sections we can write:

$$NMTF(TE1) = distance\left[min\left(NMTF_{BE_1}^{act}, \dots, NMTF_{BE_K}^{act}\right), NMTF_{BE_{K+1}}^{act}, \dots, NMTF_{BE_N}^{act}\right] \quad (9)$$

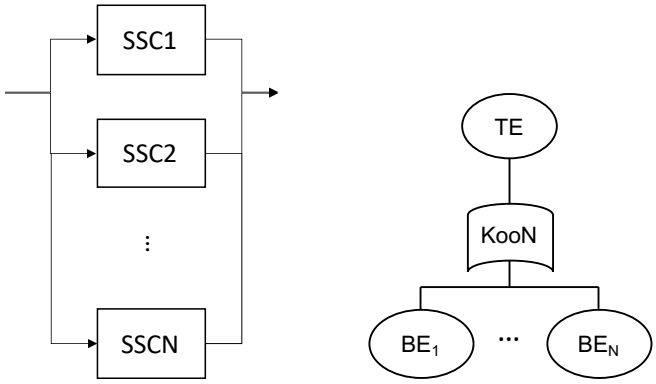


Figure 23. KooN configuration and relative Boolean logic representation.

5.4.4 NMTF Based System Reliability Calculations

In the previous sections we have indicated how it is possible to propagate NMTF based data through Boolean logic gates. Note that there is a radical difference from the propagation of probability-based data though the same gates.

An objective of this project is to model system reliability using tools/methods that are familiar to nuclear plant practitioners (e.g., use of techniques such as FTs). This section covers methods to propagate NMTF values from the basic events up to the FT top event.

Classically, reliability calculations are performed by completing these four steps:

1. Construct the FT
2. Generate the cut-sets and the MCSs from the FT
3. Assign a probability to each basic event
4. Calculate the probability of the union of the MCSs (see also Section 7)

Step 4 is typically time consuming since the probability of the union of the MCSs involve the calculation of the probability of intersection between 1, 2, 3,… MCSs as indicated in Equation (5). In Section 5.4.1, we have shown how reliability calculations based on NMTF are not based on classical set theory by, instead, on metric space (i.e., distance based) operations. Hence, exact solutions can be obtained extremely fast.

More precisely, reliability calculations using NMTF data can be performed by completing these four steps:

1. Construct the FT; at this point, a FT contains only deterministic information about the architecture of the system under consideration, i.e., it simply models how the basic events are related to each other from a functional perspective
2. Generate the cut-sets and the MCSs[d] from the FT; as also indicated in Step 1, an MCS still represents the minimal combinations of basic events which lead to the TE
3. Assign a NMTF value to each basic event
4. Calculate the NMTF of the union of the MCSs (see also Section 5.4.5) by employing Equations (7) through (9).

5.4.5 Example of System Reliability Calculation

As an example, consider the system set forth in Figure 24 which is composed of two pumps and a motor-operated control valve. Note that this is intended to be a simplified example and does not include all of the failure mechanisms or planned maintenance activities that would be applicable to the SSCs in this system.

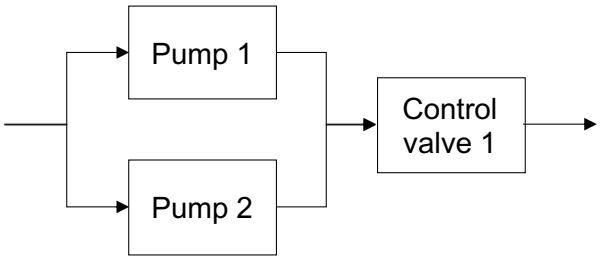


Figure 24. Simplified reliability model.

For each component, a set of failure modes have been identified and monitored as shown in Table 3.

For this system, the following notation is used:

- A: valve stress corrosion cracking
- B, D: pump bearings failure modes for pumps 1 and 2, respectively
- C, E: rotor cage winding failure for pumps 1 and 2, respectively

The FT for the considered system is shown in Figure 25 along with the NMTF for each basic event of the FT.

---

[d] Note that the generation of MCSs from the cut sets is typically performed using basic Boolean algebra rules:

- Idempotent law: $X\,X = X;\ X + X = X$
- Absorption law: $X(X + Y) = X;\ X + X\,Y = X$

When performing NMTF calculation, the generation of MCS is actually not required given formulation of Equation (8). As an example:

$$NMTF(X + X\,Y) = min\begin{Bmatrix} NMTF(X) \\ NMTF(X\,Y) \end{Bmatrix} = min\left\{ \frac{NMTF(X)}{\sqrt{NMTF(X)^2 + NMTF(Y)^2}} \right\} = NMTF(X)$$

since $NMTF(X) < \sqrt{NMTF(X)^2 + NMTF(Y)^2}$.

Table 3. ER activities and data for the system shown in Figure 24.

Component	Failure mode	Maintenance type	ER activity	NMTF data
Pumps (Includes Motor)	Pump bearings failure	PdM	Vibration monitoring coupled with prognostic and diagnostic system	Estimated RUL of pump bearings
	Rotor cage winding failure	PM	Every 5 years the rotor undergoes a rewinding process	Time from next rewinding process
Valve	Stress corrosion cracking of the valve disk failure	CBM	Measure of time to complete a full-open to full close to full-open transient	If the transient testing time falls above 4 seconds, then the component is replaced

Basic Event	NMTF
A	0.8
B	0.5
C	0.6
D	0.2
E	0.4

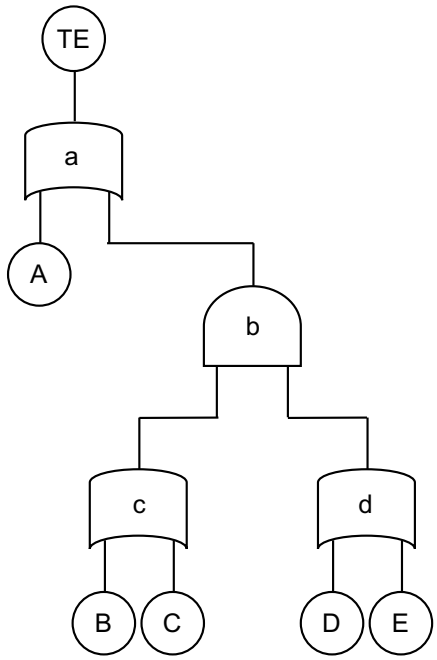


Figure 25. Reliability data expressed in terms of NMTF and reliability model for the system shown in Figure 24

The list of MCSs for this FT is follows:

$$TE = \{A, BD, BE, CD, CE\}$$

By applying the rules of Section 5.4 for the AND and OR gates:

$$NMTF(A) = 0.8$$

$$NMTF(BD) = 0.5385$$
$$NMTF(BE) = 0.5830$$
$$NMTF(CD) = 0.6324$$
$$NMTF(CE) = 0.6708$$

it is possible to determine the NMTF for the top event: $NMTF(TE) = 0.5385$.

In addition, given the continuous nature of NMTF it is possible to perform a sensitivity study for each basic event BE_j in terms of:

$$S_{BE_j} = \frac{\partial(NMTF(TE))}{\partial(NMTF(BE_j))} \tag{10}$$

Table 4 provides a summary of the sensitivities for each basic event.

Table 4. S_{BE} for the five basic events shown in Figure 47.

Failure mode	Basic event	S_{BE}
Valve stress corrosion cracking	A	0
Pump 1 bearings failure mode	B	0.93
Pump 1 motor rotor cage winding failure	C	0
Pump 2 bearings failure mode	D	0.41
Pump 1 motor rotor cage winding failure	E	0

Table 4 indicates that at the present time, the pump bearings failure modes for both pumps are the ones that dominate the NMTF of the system. Note that the definition of S_{BE} does not take into account any temporal evolution information; an alternative sensitivity analysis measure could be the following:

$$\tilde{S}_{BEj} = \frac{\frac{\partial NMTF(TE)}{\partial t}}{\frac{\partial NMTF(BEj)}{\partial t}} \tag{11}$$

5.5 Integration of ER Data into NMTF Models

One of the major challenges about probability-based reliability models is the integration of ER data. This is in particular relevant when dealing with complex components where multiple failure modes are present and each of these failure modes is addressed by a different maintenance strategy.

In Sections 5.2.1 through 5.2.4 we have shown how each maintenance strategy can be modeled using a NMTF-based language. These failure modes can be linked together depending on the architecture of the components using classical Boolean logic gates (an example is shown in Figure 26).

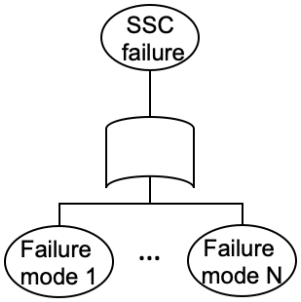


Figure 26. Reliability model of a component characterized by multiple failure modes.

In this work we are assuming that ER data is available[e]. This data can either text-based or numerical. In detail, ER data can be categorized in the following classes [6]:

1. **Operations logs.** They indicate when technical specifications action statements were entered and exited, and for particular surveillances, the exact time that the system was out-of-service. This data is relevant when performing NMTF based modeling when CBM is employed (as shown in Figure 15).
2. **Issue reports.** They provide useful information regarding the origination date, system, severity, operability, and functionality associated with plant SSCs and events. This data provides information for CM and PM maintenance options (see Figure 13 and Figure 14) since it is possible to list interval between similar occurrences and determine correlations among events.
3. **Work orders**. They provide a description of the task, start and completion dates/times, system/component, type of work (e.g., preventive maintenance) and labor hours. The labor hours are particularly useful to permit assessment of cost and economic impact.
4. **Diagnostics and prognostics data.** This data directly provides information about component status (diagnostics) and its predicted behavior (prognostics). Thus, this data can be employed to construct the MTF for CBM and PdM maintenance options (see Figure 15 and Figure 16 respectively). Here, classification and clustering algorithms [31,32] can be employed to perform (diagnostic):
    - Anomaly detection
    - Identification of failure modes/mechanisms
    - Identification of degradation status

    while stochastic models (e.g., particle filtering [33]) can be employed to evaluate (prognostics):
    - Degradation trending
    - RUL prediction

[e] The challenge at this point is the development of methods that can automatically retrieve and process ER data such that it can be automatically used by LOGOS and SR^{2}ML to manage and optimize plant resources and activities

5.6 Reliability Models for NMTF

Currently, applications of a number of reliability models are under development and evaluation (e.g., GRA models) as part of the US DOE LWRS program. These models are being constructed using standard reliability approaches that have been used in commercial nuclear power applications (i.e., ET/FT models) over many years and, as such, are based on the assumptions that each basic event is defined over a probability of failure value.

The goal now is to deploy these reliability models by feeding NMTF values at the component level and propagate these results through the models in order to determine NMTF at a level where the degraded performance or failures would result in actual consequences to plant safety or economics (e.g., the system or train level). This flow is shown in Figure 26.

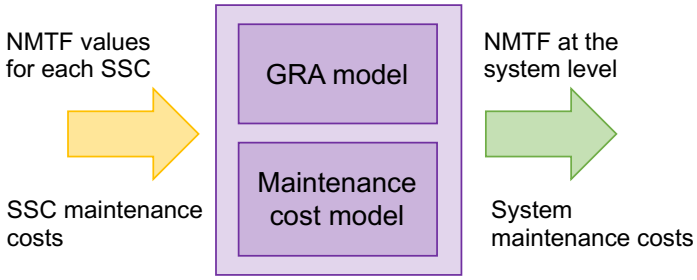


Figure 27. NMTF-based risk-informed analysis.

6. VERT FOR GRA MODELING

NPPs are currently operating in a competitive industry. As a result, NPPs need tools to increase the efficiency and reduce the costs of supplying electricity to consumers. A large focus has been placed on ER programs. The goal of ER programs is to optimize the relationship between ER and availability [6]. ER programs have proven to increase the operational performance of NPPs. However, current ER programs utilized in the nuclear sector are labor intensive and expensive.

Cost optimization is achieved when the maintenance or replacement of systems, structures, and components (SSCs) is performed just before failure occurs. There is large uncertainty when predicting the time of failure for SSCs. NPPs often take the approach of frequent inspection and maintenance to reduce the probability of SSC failure [7]. Overconservative inspection and maintenance requiring NPP derating is not a cost-effective approach to improving SSC availability. Knowing the risk imposed on NPPs from certain SSCs gives insight into lowering the frequency of inspection and maintenance activities.

Generation Risk Assessment (GRA) is a systematic method for prioritizing SSCs based on the estimated impact on the loss of future electricity generation. GRA is a useful tool in supporting ER programs but construction of the models is time consuming and costly. The Versatile Economic Risk Tool (VERT) eliminates the time and costs associated with developing the GRA model. VERT quickly and effectively evaluates the economic risk that systems and sub-systems impose on NPPs. ER programs coupled with VERT can improve operational performance. Knowing what systems and sub-systems contribute the most to derates at NPPs can give valuable insight to maintenance and inspection activities. Improvement of NPP operational performance also can directly result in safety improvements and reduced risks [8].

Understanding the degradation of SSCs has become vital for establishing long active periods of NPP operation [4]. Several programs, including the proactive management of materials degradation (PMMD)

program developed by the NRC, have been undertaken to address the SSC degradation. VERT introduces time-dependent capabilities into GRA modeling. VERT can directly use the knowledge gained from degradation programs to provide time-dependent analysis of SSC economic risks.

More details about VERT are shown in Appendices A, B, and C.

6.1 Methodology

NPPs have been assessing safety risks effectively with Probabilistic Risk Assessment (PRA) for decades. The GRA methodology is largely based on PRA concepts, focusing on what can go wrong, how likely it is, and what the consequences are. NPP shutdowns and reductions of power (derates) are the only scenarios identified in GRA. The components evaluated in GRA are identified using two of the criteria in the INPO AP-913 Scoping and Identification of Critical Components section in which any component failure resulting in a significant power derate or shutdown is considered a critical component [5]. GRA risk is the product of critical components' failure likelihoods and the consequences of the failures. VERT compresses GRA risks into a single program. GRA has not yet been adopted by industry because of the large time and manpower resource constraints (in particular use of personnel knowledgeable in PRA methods and tools). VERT addresses the time requirements and reduces the required manpower and specialized knowledge to use the program.

Risk is assumed to be constant with respect to time in most GRA models. However, critical components are subject to aging and degradation which increases failure probabilities over time. The integration of degradation models into GRA provides more realistic risk estimates. Predicting how economic losses change as equipment in NPPs degrade is imperative for making long-term business decisions [6].

6.2 VERT Architecture

The program developed is a generic economic risk tool adaptable to any Light Water Reactor (LWR) NPP. The foundation of the program is based on equipment found in all LWR power units. Figure 28 shows the constituents of VERT at the highest systemic level. The four main sectors comprise the critical components in NPPs.

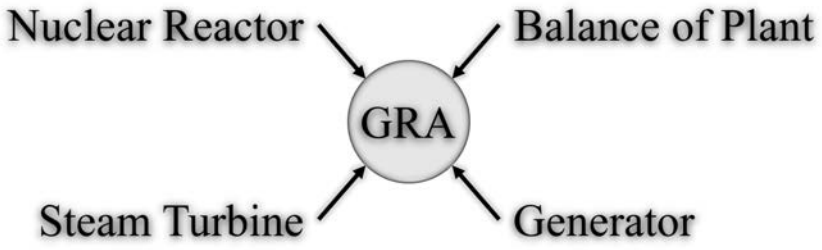


Figure 28. The four highest systems of VERT.

The main systems are further broken down into the comprising sub-systems. Figure 29 shows the balance of plant composition of sub-systems as an example.

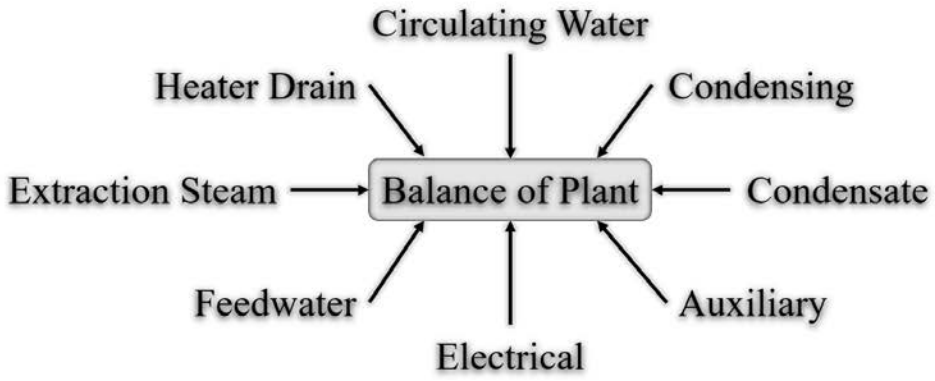


Figure 29. Balance of plant system constituents.

Each sub-system shown in Figure 29 contains all of the equipment necessary for power generation. Appendix A shows the constituents of the four main systems.

VERT couples INL's RAVEN and SAPHIRE software packages with a specified database to perform time-dependent economic analyses. The general flow of VERT is shown in Figure 30. The database and the model in SAPHIRE are interconnected with component specified inputs. The SAPHIRE model contains all the components in the database and the respective location in the NPP. The database is a comma-separated values (.CSV) spreadsheet with all the unavailability information of the components in the SAPHIRE model. VERT generates all the necessary post-processing actions and file creations necessary to couple RAVEN with the data spreadsheet and SAPHIRE. After VERT executes, RAVEN can run to perform the analysis and produce the results.

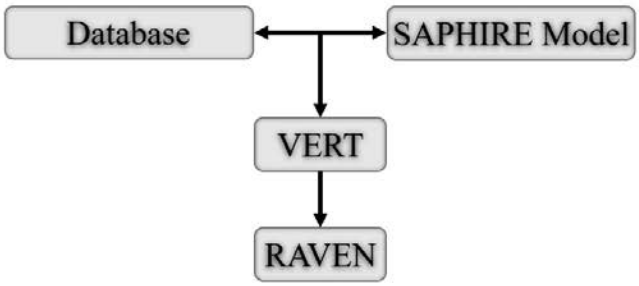


Figure 30. Flow chart of VERT.

VERT uses the SAPHIRE modeling software to evaluate the probability of NPP tripping or derating. FTs are used to represent systemic logic to obtain cut sets for the power reduction events. A post-processing tool available with SAPHIRE is used to multiply the cut set probabilities by the consequences of each cut set. The estimated risk is obtained with the equations used for GRA [7]:

$$Risk = Likelihood \times Consequences \tag{12}$$

$$Consequences = Derate(\%) \times Rated\ Capacity(MW) \times (MTTR + Time\ to\ restore\ power)(hr) \times Cost\left(\frac{\$}{MWh}\right) \quad (13)$$

VERT is adaptable to any NPP using generic Pressurized Water Reactor (PWR) and Boiling Water Reactor (BWR) databases and FT structures. The program also enables the user to add or remove the quantity of trains in any system, e.g. a four loop PWR would have four steam generator trains. The user may also change the failure probabilities and consequences of equipment failure to match the particular NPP design and operating data. The default failure rates, MTTR, and derate percentages are derived from the North American Electric Reliability Corporation – Generating Availability Data System (NERC-GADS). GADS is discussed further in the appendices.

VERT can integrate degradation into the economic models. Equations 14 and 15 represent linear and exponential degradation of the failure rates of the components. The user can specify the equation type for each component and the respective "b" value.

$$\lambda(t) = \lambda_0(1 + bt) \quad (14)$$

$$\lambda(t) = \lambda_0 e^{bt} \quad (15)$$

6.3 Results

VERT quantifies the risk of the NPP system specified by the user and automatically generates several plots and spreadsheets. The spreadsheets give estimates of economic loss imposed on the NPP according to specific sections of the NPP. The plots display relationships between the systems and sub-systems of the NPP. Figure 31 shows an example of the total system contribution to economic loss and the individual contributions by the main constituent systems of the NPP relative to one another over a projected 10 year period.

The results shown in the plot are also generated for the sub-systems comprising the Balance of Plant, Generator, Reactor, and Steam Turbine systems. The plots provide insights into what systems contribute the most to economic loss as well as the least. If a utility used VERT and wished to investigate strategies to improve the economic risk of the reactor system, for example, Figure 32could be utilized to identify the risk in the reactor sub-systems relative to each another.

The plot shows the sub-systems corresponding to cause codes[f] 2200-2399 (reactor coolant system) and 2400-2599 (e.g., steam generators and steam system) contribute the most to the risk of lost generation for a four loop NPP. Maintenance and inspection activities could be used to improve the economic risk of these sub-systems. Identifying the lowest contributors to risk will also provide insight into what maintenance and inspection activities could be decreased or defunded. Time and money spent on the sub-systems contributing the least amount to risk could be re-distributed to those systems that provide greater contribution to risk (e.g., the reactor coolant system, steam generators, and steam system).

[f] These codes match the NERC-GADS ones.

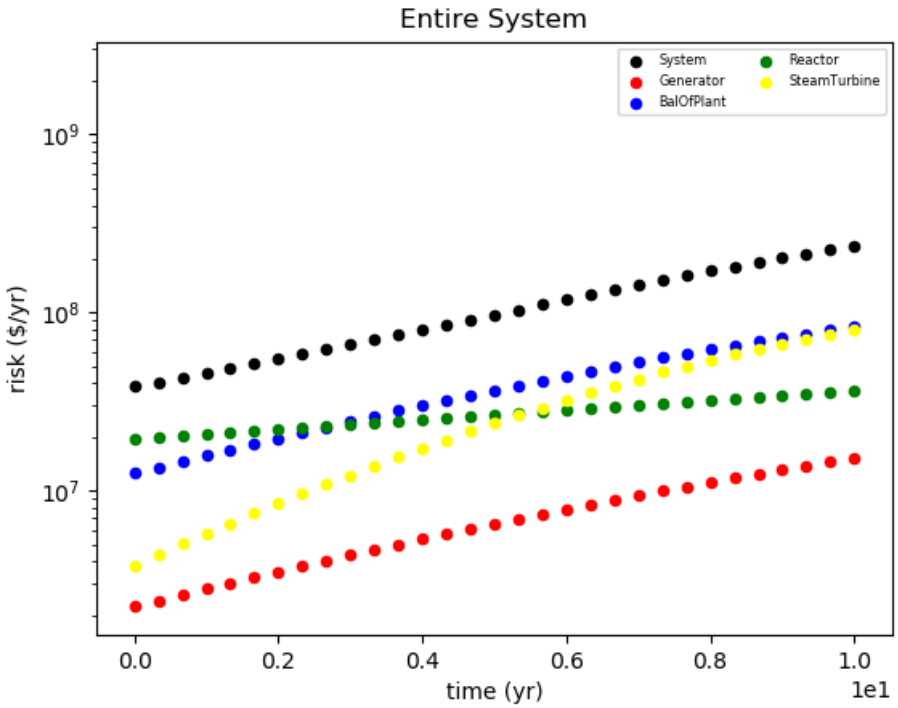


Figure 31. Logarithmic plot of the NPP system relative risk estimates.

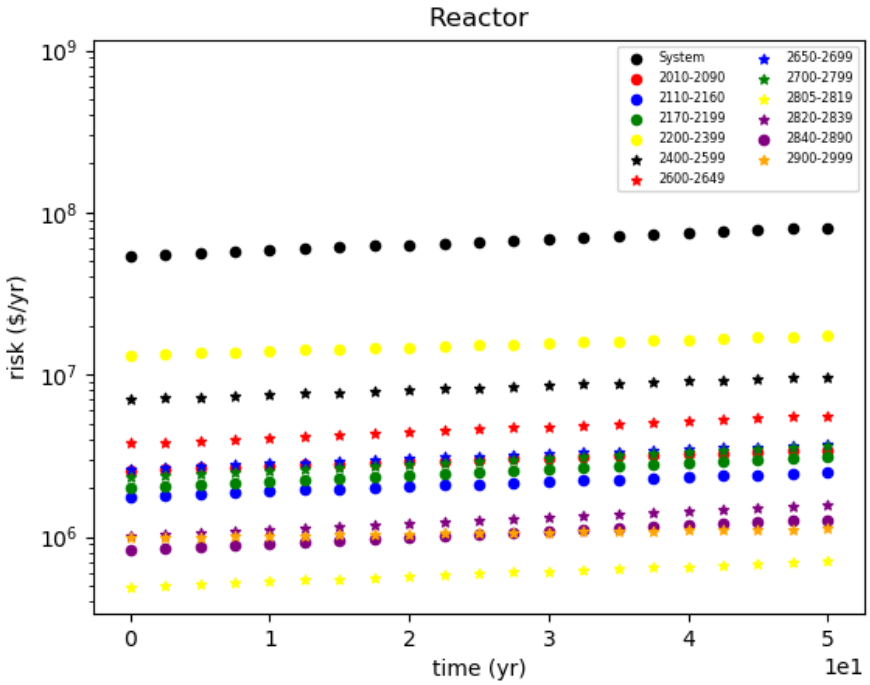


Figure 32. Reactor system economic risk plot.

6.4 Improvements

The risk from the individual basic events is not directly quantified with VERT. The risk of small groups of components is quantified, but there is no further breakdown of the components in the group. The data spreadsheet contains a row with the associated risk for each individual basic event. A future advancement to assist in comparing basic events to one another is the automatic generation of bar charts with the basic events and the associated risks.

The uncertainty sampling method with RAVEN increases the computing time substantially. The uncertainty sampling with SAPHIRE currently works for most FTs. The uncertainty sampling is absent for four of the FTs. The uncertainty will be applicable to all the FTs and plots will be automatically generated for VERT results.

Maintenance and planned derating occur during power generation operations. Knowing the unavailability of equipment due to failures and planned unavailability will provide information to improve production. The pc-GAR software holds information on maintenance unavailability and can be integrated with VERT.

Versions of VERT compatible with Linux and Mac operating systems need to be developed. Currently, VERT is compatible with Windows systems with SAPHIRE and RAVEN. Use with Linux and Mac operating systems have not been tested.

There are multiple importance measures relating to NPP performance. Unplanned capability loss factor, thermal performance, chemistry performance, safety system performance, etc. are all statistics valuable to understanding how safely and economically an NPP is performing [8]. Some of these measures, such as the unplanned capability loss factor, can directly be generated from VERT. Some modifications in the coding of VERT can automatically quantify some of these importance measures.

7. MCS SOLVER MODEL

The reliability model at the system level typically consists of a hierarchical set of FTs, which are made up of a Boolean logic structure designed to link the Top Event (TE) under consideration (e.g., system failure to perform its desired function) given the Boolean status of its basic components, the BEs.

The FT structure can be solved in probability terms with the goal of determining the probability of TE occurrence, given the probability values associated to each BE. This can be achieved by first generating a list containing combinations of BEs that lead to the TE under consideration (i.e., the minimal cut sets [MCSs]). Then, the probability associated with the TE can be calculates as the probability of the union of all MCSs.

During FY-20, we have developed a model designed to determine TE probability provided the list of MCSs and the probability associated to each BE—the `MCSSolver` model, which can be directly employed within RAVEN and linked to component reliability models. Below, we indicate how the computation of the TE probability is performed. The following notation is used:

- Ξ : Set of MCSs, $MCS_n \in \Xi \ \ (n = 1, \dots, N)$
- Ψ : Set of BEs, $BE_l \in \Psi \ \ (l = 1, \dots, L)$

Note that:

- $MCS_n \subset \Psi$, i.e., MCS_n can be considered a set of BEs.
- $MCS_p \nsubseteq MCS_q$

- It can happen that $MCS_p \cap MCS_q \neq \emptyset$

From a model perspective, the `MCSSolver` model input-output variables are the following

1. Inputs variables: BE probability $P(BE_l)\ \ l = 1, \dots, L$
2. Output variables: TE probability $P(TE)$

The `MCSSolver` is initialized by reading from the list of MCSs from the file and it constructs the terms required to determine the probability of TE, $P(TE)$, as follows:

$$P(TE) = P\left(\bigcup_{n=1}^{N} MCS_n\right) = \\ = \sum_{n=1}^{N} P(MCS_n) - \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} P(MCS_n\ MCS_m) + \cdots \\ + (-1)^{N-1}\, P(MCS_1 \dots\ MCS_N) \tag{16}$$

where

$$P(MCS_n) = \prod_{BE \in MCS_n} P(BE)$$

$$P\left(MCS_p\ MCS_q\right) = \prod_{BE \in (MCS_p\ \cup\ MCS_q)} P(BE) \tag{17}$$

$$P\left(MCS_p\ MCS_q MCS_r\right) = \prod_{BE \in (MCS_p \cup MCS_q \cup MCS_r)} P(BE)$$

The analysis can specify the parameter *solverOrder* which specifies the maximum calculation envelope for $P(TE)$, i.e., the maximum number of MCSs to be considered when evaluating the probability of their union.

8. SR²ML REPOSITORY

SR²ML is a software package that contains a set of reliability models designed to be interfaced with the INL developed RAVEN code. These models can be employed to perform both static and dynamic system risk analysis and determine the risk importance of specific elements of the considered system. These developed models are stored in the INL GitLab software repository as shown in Figure 33.

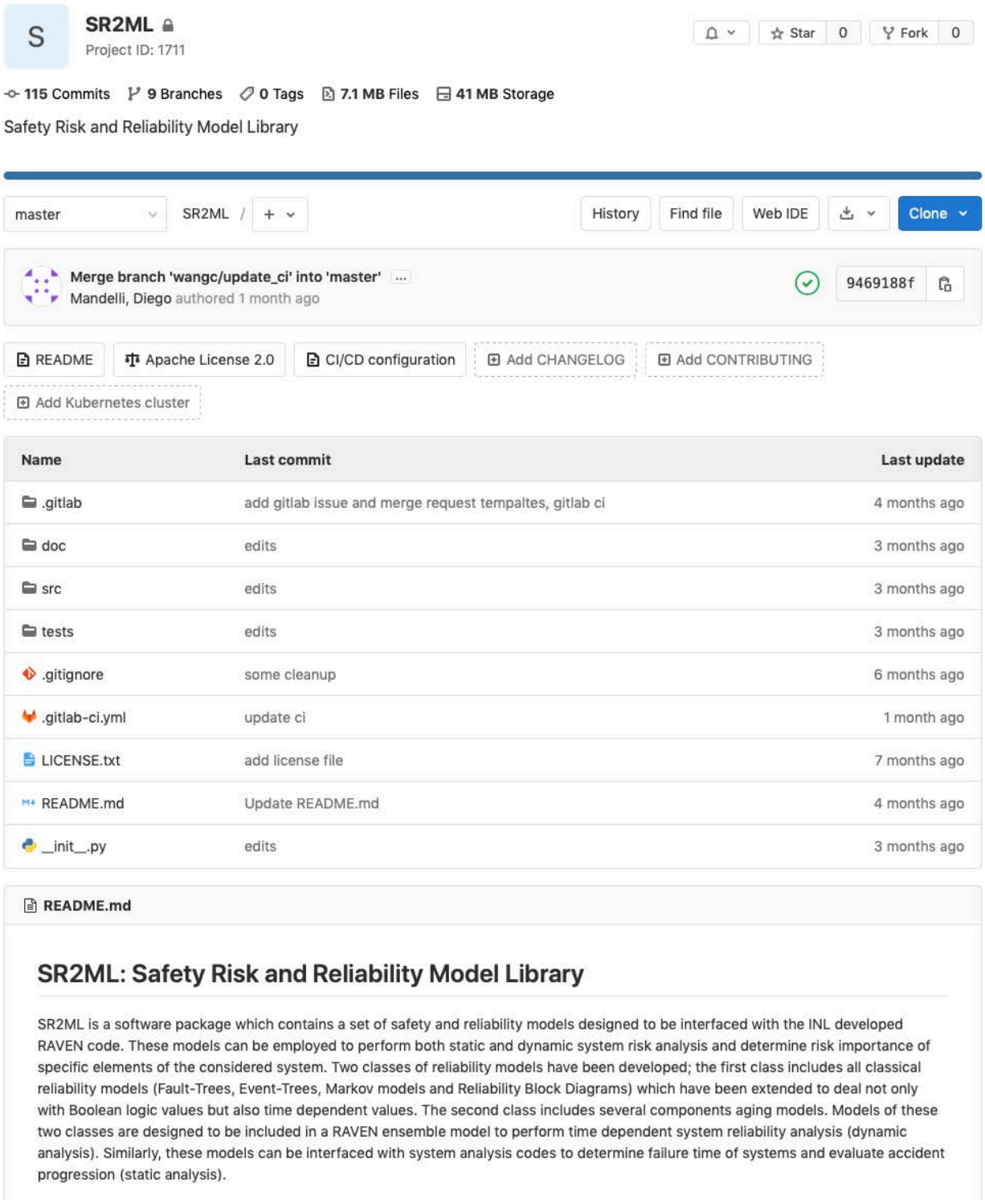


Figure 33. Snapshot of the SR^{2}ML repository.

Two classes of reliability models have been developed; the first class includes all classical reliability models (FTs, ETs, Markov models and reliability block diagrams) which have been extended to deal not

only with Boolean logic values but also time dependent values. The second class includes several component ageing models. Models of these two classes are designed to be included in a RAVEN ensemble model to perform time dependent system reliability analysis (i.e., dynamic analysis). Similarly, these models can be interfaced with system analysis codes within RAVEN to determine the failure time of systems and evaluate accident progression (i.e., static analysis).

Reliability models/functions are the most frequently used in life data analysis and reliability engineering. These models/functions give the probability of a component operating for a certain amount of time without failure. As such, the reliability models are functions of time, in that every reliability value has an associated time value. In other words, one must specify a time value with the desired reliability value. This degree of flexibility makes the reliability model a much better reliability specification than the mean time to failure (MTTF), which represents only one point along the entire reliability model.

8.1 Probabilistic Modeling

From probability and statistics, given a continuous random variable x, we denote [23]:

- The probability density function (pdf), as $f(x)$.
- The cumulative distribution function (cdf), as $F(x)$.

If x is a continuous random variable, then the probability that x takes on a value in the interval $[a, b]$ is the area under the pdf $f(x)$ from a to b:

$$P(a \leq x \leq b) = \int_a^b f(x)dx \tag{18}$$

The cumulative distribution function is a function $F(x)$ of a random variable x, and is defined for a number x_0 by:

$$F(x_0) = P(x \leq x_0) = \int_{-\infty}^{x_0} f(x)dx \tag{19}$$

That is, for a given value of x_0, $F(x_0)$ is the probability that the observed value of x would be, at most, x_0. The mathematical relationship between the pdf and the cdf is given by:

$$F(x) = \int_{-\infty}^{x} f(s)ds \tag{20}$$

Conversely:

$$f(x) = \frac{dF(x)}{dx} \tag{21}$$

The functions most used in reliability engineering and life data analysis, namely the reliability functions and failure rate functions, can be determined directly from the pdf definition, or $f(t)$. Different distributions exist, such as Lognormal, Exponential, Weibull etc., and each of them has a predefined pdf $f(t)$. These distributions were formulated by statisticians, mathematicians and engineers to mathematically model or represent certain behavior. Some distributions tend to better represent life data and are most are referred to as lifetime distribution.

8.2 Reliability and Failure Rate Models

Given the mathematical representation of a distribution, we can derive all functions needed for reliability analysis, i.e. reliability models/functions. This would only depend on the value of t after the value of the distribution parameters are estimated from the data. Now, let t be the random variable defining the lifetime of the component with cdf $F(t)$, which is the time the component would operate before failure. Note that the following represent standard relations in statistics; for the further information the reader should consult standard statistics texts. The cdf $F(t)$ of the random variable t is given by [23]:

$$F(t) = P(s \leq t) = \int_{-\infty}^{t} f(s)ds \tag{22}$$

If $F(t)$ is a differentiable function, then the pdf of t is given by:

$$f(t) = \frac{dF(t)}{dt} \tag{23}$$

The reliability function or survival function $R(t)$ of the component is given by:

$$R(t) = P(s > t) = 1 - P(s \leq t) = 1 - F(t) \tag{24}$$

This is the probability that the component will operate after time t, sometimes called survival probability.

The failure rate of a system during the interval $[t, t + \Delta t]$ is the probability that a failure per unit time occurs in the interval, given that a failure has not occurred prior to t, the beginning of the interval. The failure rate function (i.e., instantaneous failure rate, conditional failure rate) or the hazard function is defined as the limit of the failure rate as the interval approaches zero. Hence [23]:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t R(t)} = \frac{1}{R(t)} \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{1}{R(t)} \frac{dF(t)}{dt} = \frac{f(t)}{R(t)} \tag{25}$$

The failure rate function is the rate of change of the conditional probability of failure at time t. It measures the likelihood that a component that has operated up until time t fails in the next instant of time. Generally, $\lambda(t)$ is the one tabulated because it is measured experimentally and because it tends to vary less rapidly with time than the other parameters. When $\lambda(t)$ is given, all other three parameters $F(t), f(t), R(t)$ can be computed as follows [23]:

$$R(t) = \exp\left(-\int_0^t \lambda(s)ds\right) \tag{26}$$

$$f(t) = \lambda(t)R(t) = \lambda(t) \exp\left(-\int_0^t \lambda(s)ds\right) \tag{27}$$

$$F(t) = 1 - R(t) = 1 - \exp\left(-\int_0^t \lambda(s)ds\right) \tag{28}$$

The Mean Time Between Failure (MTBF) can be obtained by finding the expected value of the random variable t, time to failure. Hence:

$$MTBF = \int_0^\infty t f(t)dt = \int_0^\infty R(t)dt \tag{29}$$

8.3 Ageing Models

We have implemented several of the most useful reliability models based on different probability distributions for describing failure of continuous operating devices within the SR^{2}ML tool, including: Exponential, Erlangian, Gamma, Lognormal, Fatigue Life, Weibull, Exponential Weibull, Bathtub, Power Law model, and Log Linear model. These models are described in further detail in the sections that follow. These represent standard statistical distributions used in reliability engineering; for the further information the reader should consult standard statistics and reliability engineering texts [23].

The Lognormal Model

The probability density function of the lognormal is given by

$$f(t) = \frac{1}{\alpha t\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln\frac{t-\mu}{\beta}}{\alpha}\right)^2} \tag{30}$$

where $t \geq \mu, \mu, \alpha, \beta > 0$, and β is the scale parameter, α is the shape parameter, and μ is the location parameter.

The Exponential Model

The exponential distribution can be used in reliability as a model of the time to failure of a component, and its probability density function is given by

$$f(t) = \lambda e^{-\lambda(t-\mu)}, t \geq 0 \tag{31}$$

where $t \geq \mu, \mu, \lambda > 0$, λ is the mean failure rate or the inverse of the scale parameter, and μ is the location parameter.

The Weibull Model

The probability density function of the three-parameter Weibull distribution is given by:

$$f(t) = \frac{\alpha}{\beta}\left(\frac{t-\mu}{\beta}\right)^{\beta-1} e^{-\left(\frac{t-\mu}{\beta}\right)^{\alpha}} \tag{32}$$

where $t \geq \mu, \mu, \beta, \alpha > 0$, and β is the scale parameter, α is the shape parameter, and μ is the location parameter.

The Erlangian Model

The probability density function of the Erlangian distribution is given by:

$$f(t) = \frac{\lambda\big(\lambda(t-\mu)\big)^{k-1}\exp\big(-\lambda(t-\mu)\big)}{(k-1)!} \tag{33}$$

where $t \geq \mu, \mu, \lambda > 0$, and λ is the inverse of the scale parameter, k is a positive integer that controls the shape, and μ is the location parameter.

The Gamma Model

The probability density function of the Gamma is given by:

$$f(t) = \frac{\beta\big(\beta(t-\mu)\big)^{\alpha-1}\exp\big(-\beta(t-\mu)\big)}{\Gamma(\alpha)} \tag{34}$$

where $t \geq \mu, \mu, \beta, \alpha > 0$, and β is the inverse of scale parameter, α is the shape parameter, and μ is the location parameter.

The Fatigue Life Model (Birnbaum-Saunders)

The probability density function of the Fatigue Life model is given by:

$$f(t) = \frac{\frac{t-\mu}{\beta}+1}{2\alpha\sqrt{2\pi\left(\frac{t-\mu}{\beta}\right)^{3}}}\exp\left(-\frac{\left(\frac{t-\mu}{\beta}-1\right)^{2}}{2\left(\frac{t-\mu}{\beta}\right)\alpha^{2}}\right) \tag{35}$$

where $t \geq \mu, \mu, \beta, \alpha > 0$, and β is the scale parameter, α is the shape parameter, and μ is the location parameter.

The Exponentiated Weibull Model

The probability density function of the Exponentiated Weibull distribution is given by:

$$f(t) = \gamma\alpha\left(1-\exp\left(-\left(\frac{t-\mu}{\beta}\right)^{\alpha}\right)\right)^{\gamma-1}\left(\frac{t-\mu}{\beta}\right)^{\alpha-1}\exp\left(-\left(\frac{t-\mu}{\beta}\right)^{\alpha}\right) \tag{36}$$

where $t \geq \mu, \mu, \beta, \alpha, \gamma > 0$, and β is the scale parameter, α and γ is the shape parameter, and μ is the location parameter.

The Bathtub Model

The probability density function of the Bathtub model is given by:

$$f(t) = \exp\left(-c\beta\left(\frac{t-\mu}{\beta}\right)^{\alpha} - (1-c)\left(\exp\left(-\left(\frac{t-\mu}{\theta}\right)^{\rho}\right)-1\right)\right) \tag{37}$$

where $t \geq \mu, \mu, \beta, \alpha, \theta, \rho, c > 0$, and β, θ is the scale parameter, α, ρ are shape parameters, $c \in [0,1]$ is the weight parameter, and μ is the location parameter.

The Power Law Model

The hazard rate satisfies a power law as a function of time:

$$\lambda(t) = \lambda + \alpha(t-\mu)^{\beta} \tag{38}$$

where $t \geq \mu$, $\mu, \alpha, \beta, \lambda > 0$, and μ is the location parameter.

The Log Linear Model

The hazard rate satisfies an exponential law as a function of time:

$$\lambda(t) = \exp\big(\alpha + \beta(t-\mu)\big) \tag{39}$$

where $t \geq \mu$, $\mu, \alpha, \beta > 0$, and μ is the location parameter.

8.4 Maintenance Models

Maintenance models are models designed to model maintenance and testing from a reliability perspective. These models are designed to optimize preventive maintenance at the system level. Two classes of models are considered in this report: operating and stand-by.

For an operating component, the unavailability $\tilde{u}_i$ is calculated as:

$$\tilde{u}_i = \frac{\lambda\, T_i^{rep}}{1 + \lambda\, T_i^{rep}} + \frac{T_i^{DT}}{T_i^{PM}} \tag{40}$$

where:

- T_i^{rep}: MTTR
- λ: component failure rate
- T_i^{DT}: mean time to perform PM
- T_i^{PM}: PM interval

For a stand-by component, the unavailability $\tilde{u}_i$ is calculated as:

$$\tilde{u}_i = \rho + \frac{1}{2}\lambda\, T_i^{test} + \frac{T_i^{dt}}{T_i^{test}} + \big(\rho + \lambda\, T_i^{test}\big)\frac{(T_i^{rep})}{T_i^{test}} + \frac{T_i^{DT}}{T_i^{PM}} \tag{41}$$

where:

- T_i^{rep}: MTTR
- λ: component failure rate
- T_i^{DT}: mean time to perform PM
- T_i^{PM}: PM interval
- T_i^{dt}: mean time to perform surveillance test
- T_i^{test}: surveillance test interval

In order to include component ageing in this analysis, the component failure rate λ_i is not constant but can change with time such that $\lambda_i = \lambda_i(t)$.

9. RAVEN ENSEMBLE AND LOGICAL MODELS

RAVEN's scope [22] is to provide a set of capabilities to build analysis flows based on UQ, reliability analysis, optimization and data analysis techniques to be applied to any physical model(s). The main objective of RAVEN is to assist the engineer/user to perform:

- Uncertainty quantification
- Probabilistic risk and reliability analysis
- Data mining analysis
- Model optimization

The RAVEN software employs several novel and unique techniques, based on extensive use of artificial intelligence algorithms, such as adaptive (smart) sampling, adaptive branching algorithms (dynamic event tree), time-dependent statistical analysis and data mining. The overall set of algorithms implemented in the RAVEN software are designed to handle highly non-linear systems, characterized by system response discontinuities and discrete variables. These capabilities are crucial for handling complex system models, such as those used in the analyses of NPPs. For example, reliability surface analysis, as implemented in RAVEN, is unique and capable to handle non-linear, discontinuous systems, allowing for faster and more accurate assessing of failure risk for complex systems.

Among its different capabilities, RAVEN provides the unique functionality to combine any model (e.g., physical models, surrogate models, data analysis models, etc.) in a single entity called the Ensemble Model, where each model can feedback into others. In the following section, a more detailed description of this capability is reported.

9.1 Ensemble Model

In several cases multiple models need to be interfaced with each other since the initial conditions of some models are dependent on the outcomes of others. To solve this problem, RAVEN provides a model entity named `EnsembleModel`. This class is able to assemble multiple models of other categories (i.e., Code, External Model, Reduced Order Models - ROM), identifying the input/output connections, and, consequentially the order of execution and which sub-models can be executed in parallel.

Figure 34 shows an example of an `EnsembleModel` that is constituted of 3 sub-models (e.g., Reduced Order Models [13], Codes, or External Models) where:

- *Model 2* is connected with *Model 1* through the variable Θ (Model 1 output and Model 2 input);
- *Model 3* is connected with *Model 2* through the variable Π (Model 2 output and Model 3 input);

In this case, the `EnsembleModel` is going to drive the execution of all the sub-models in sequence, since each model (except *Model 1*) is dependent on the outcomes of previously executed models.

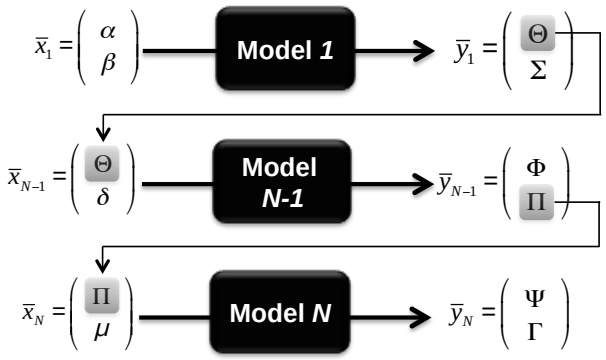


Figure 34. Example of an `EnsembleModel` constituted of 3 sequential sub-models.

9.2 Logical Model

Logical modeling in RAVEN is a process where multiple models with the same inputs and outputs are grouped together and controlled by a user provided control function to compute the responses of interest. A model entity named `LogicalModel` is developed and implemented in RAVEN to group multiple models, i.e. codes, external models, reduced order models and ensemble models, and identify which model to execute through the control function. A logical system consists of four different types of models connected via a control function as illustrated in Figure 35, and a detailed example using `LogicalModel` to group different failure mechanisms of pumps and valves is depicted in Figure 36 and is used for the case studies in this report.

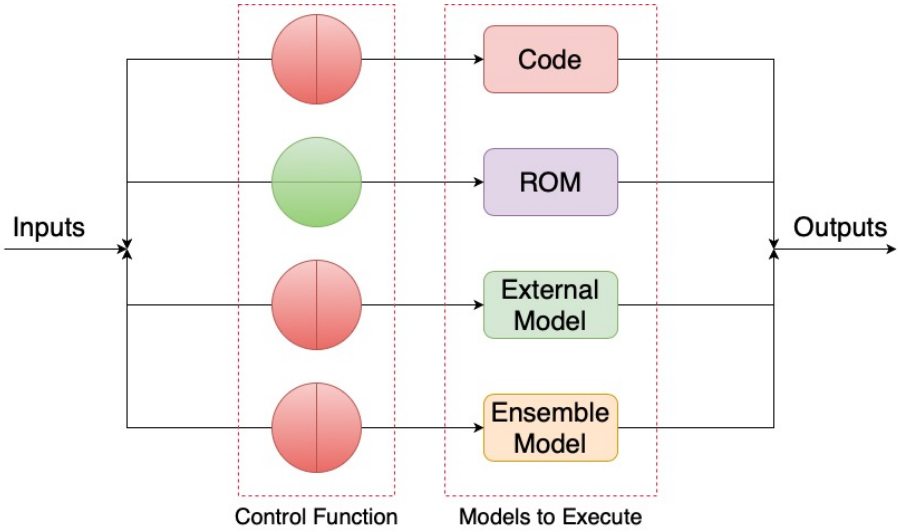


Figure 35. Illustration of `LogicalModel` constituted by four different types of RAVEN models.

```
<LogicalModel name="V1" subType="">
  <Model class="Models" type="ExternalModel">valveTypeA</Model>
  <Model class="Models" type="ExternalModel">valveTypeB</Model>
  <Model class="Models" type="ExternalModel">valveTypeC</Model>
  <ControlFunction class="Functions" type="External">control_V1</ControlFunction>
  <alias variable='cdf_F_V1' type='output'>cdf_F</alias>
</LogicalModel>

<LogicalModel name="P1" subType="">
  <Model class="Models" type="ExternalModel">pumpTypeA</Model>
  <Model class="Models" type="ExternalModel">pumpTypeB</Model>
  <Model class="Models" type="ExternalModel">pumpTypeC</Model>
  <ControlFunction class="Functions" type="External">control_P1</ControlFunction>
  <alias variable='cdf_F_P1' type='output'>cdf_F</alias>
</LogicalModel>
```

Figure 36. Example of `LogicalModel` modeling pump and valve with different types of failure mechanisms.

10. DATA, MODELING, AND FORECASTING FOR NUCLEAR PLANT SYSTEM HEALTH

The goal of this research is to provide methods and computational tools to integrate operational data and simulation models to forecast the degradation of SSCs for prognostic health management.

A considerable expense for nuclear power is O&M activities, and yet the health of SSCs is currently determined by regular time-based maintenance and inspection. Plants could better manage their O&M costs if they had a better understanding of the health of components and could forecast when they need maintenance or replacement. Such forecasts could estimate for SCCs the time to failure or the remaining useful life. In addition, a major concern for nuclear utilities is maintaining and replacing high-value assets. High value assets could include components of the reactor internals (e.g., control rod guide tubes, baffle-former bolts, thermal shield flexures), reactor coolant pumps (e.g., seals, electrical motors), or branch line piping. Such assets are often critical to the operation of the plant and have been shown to impact outage schedules and maintenance costs. For example, evidence of unexpected fatigue accumulation in a pipe junction or discovering within the reactor internals unacceptable guide card wear or baffle-former bolt degradation could require extended outages. The ability to estimate the condition and life expectancy of critical assets is essential for planning outages, O&M activities, the purchase and installation of replacements, and risk-informed systems assessment. For this, a critical need is improved assessment of the health of SSCs.

In this research, we have developed the signal processing and prognostics approach necessary to evaluate and assess SSC health. This health assessment provides much needed input to risk-informed systems assessment. While there is currently a lot of interest in purely data-driven approaches, for nuclear power, faults and failures of components are in fact quite rare. When there arelots of data, data-driven methods, like machine learning can be used. When there is little or no data, model-based methods that rely on first principles are necessary. Nuclear power lies somewhere in between, and what is needed is an approach that properly combines model-based and data-driven techniques. Such hybrid techniques should use the best of both approaches to model components and estimate their health. To meet this need, we have explored techniques that provide some estimate SSC health.

In Appendix E, we have developed the signal processing and prognostics approach that can be used to assess health. In contrast to purely data-driven approaches, this approach properly combines model-based and data-driven techniques to evaluate unknown characteristics in a system.

Of particular interest for this project was the degradation of reactor internals during the life of a reactor and from outage to outage. Due to the large flow velocities in a reactor, there can be considerable vibration of the reactor internals. In this research, we are specifically interested in the core barrel and its support structures. The vibrations of the reactor internals are measured by fluctuations in the ex-core neutron detectors, which presents its own challenges since this is an indirect measure of the vibrations. Over time, the nature and contact between the reactor internals and the reactor vessel changes. A gradual decline in contact frequency as the unit ages is a result of a reduction in contact and an increase in the space between the vessel and internals, e.g., wear in the radial key.

Purely data-driven approaches for component health use historical values for time to failure and establish inspection and maintenance activities based on that history. Other than periodic inspections, little about the operating condition of the component is used to establish its health. Statistical characteristics of signals to establish residuals can be tracked and changing trends can be used to signal the need for inspection, repair, or replacement. But such approaches are missing knowledge about the physical characteristics of the component and cannot provide any physical insight to the degradation mechanism.

Machine learning can be used to identify patterns and predict system behavior based on historical measurements. It is therefore natural to expect that machine learning techniques can be applied to this problem. But machine learning strategies cannot be applied in a "black box" manner. There are two reasons for this. First, the measurements are do not provide a direct class or label about the state of the reactor internals. Therefore, the signals may not allow an algorithm to find a relationship between the degradation of the radial key, say, because the data do not include any measurements inside the reactor vessel. The neutron noise data consist of zero observations inside the reactor vessel and so no input to output relationships can be established. Second, physical context must be provided by numerical models. A numerical model makes use of the underlying governing equations, physical laws, and Subject Matter Expert (SME) engineering insight. If the state of the system is known, the dynamics can be simulated, and its response can be compared to the measurements. This inverse structure — given a response then learn the input — is aligned with the goal of determining the characteristic of the contact.

The challenge when integrating model-based and data-driven techniques is that there must be a coherent framework that combines the known physics-based model and the as yet unknown data-driven model of the degradation mechanism. In the reactor internals case studied here, what must be captured is the interaction between the vibrational model of the reactor internals and a data-driven model that describes the contact mechanics. This interaction is a feedback mechanism where the vibrations drive the contact mechanics and contact then affects vibration. We have modeled explicitly the feedback interaction between the vibrational system and the as-yet unknown contact mechanics.

We have used a full Bayesian approach for inferring the unknown contact model by combining the neutron noise measurements with computer simulations. The computer simulations provide the context, and the full Bayesian approach represents the unknown states as probability distributions. In an operational context, decisions can thus be made by considering the uncertainty in the state and the type of action that should be scheduled. For example, if the expected behavior is considered acceptable but the uncertainty is "large" an inspection could be scheduled. However, if the behavior is considered to be "precisely" near off-nominal conditions, then a replacement can be ordered and scheduled before a planned outage.

By tracking the contact mechanism over time, we can track changes in the health of the true system. More importantly, once trained we can separate the contact model from the vibration model. We can use the contact model to elucidate the dependence of contact on operating condition. This provides new insight to the character of the contact in a way that a priori contact models cannot. This contact model can then be

used for further hypothesis testing of the physics of contact and for forecasting health of components in real-time using real data.

11. CONCLUSIONS

In this report we have shown the recent developments in the PHM project and how it relates to its counterpart, the RIAM project. During FY-20 the PHM project focused on providing a large variety of reliability models at both the component and system level which can be easily linked to each other through the RAVEN analysis platform.

The models at the component level are designed to include maintenance, testing, and ageing effects using a probability-based language. The models at the system level are designed to be adaptable to most U.S. plant configurations (e.g., PWRs and BWRs) using well known FT based reliability structures. Component models can be linked to system reliability models such that it is possible to propagate maintenance, testing and ageing effects at the system level using a probability-based language. This language can be changed into a new one which better reflects the types of data available from plant ER programs and the M&D center. Rather than inferring complex probabilistic models from limited amounts of data, we change to the language which better suits the data: the margin to failure. By doing so we are able not only to better utilize the available data, but we can still employ the developed reliability models with different solvers compatible with the concept of MTF.

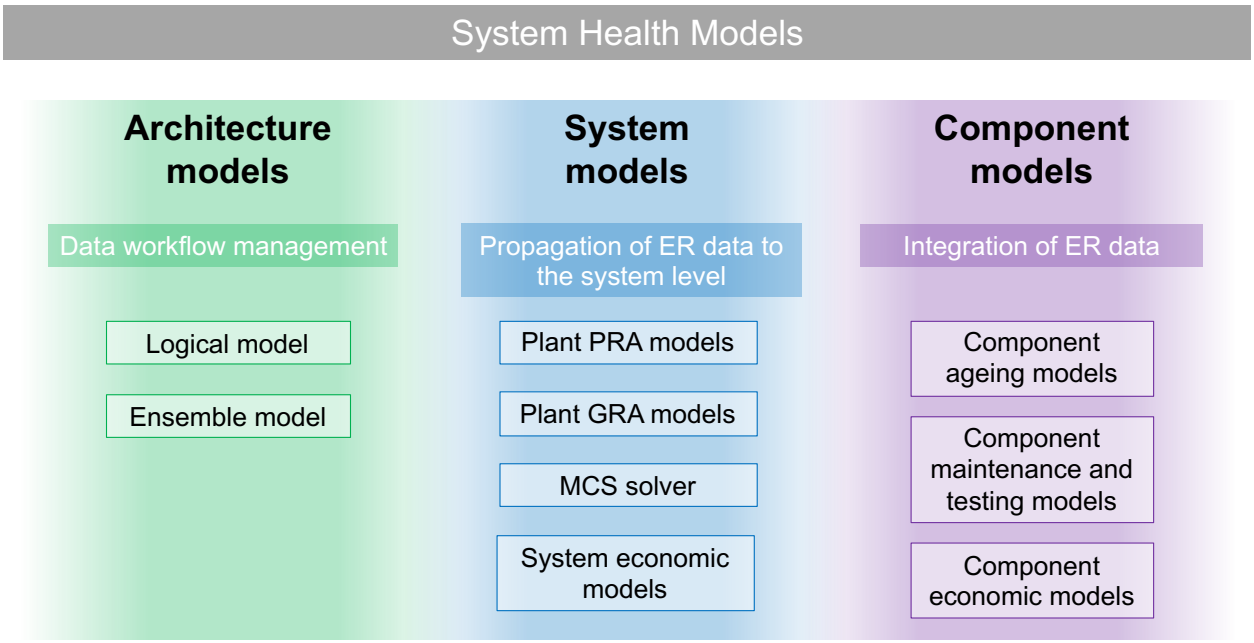


Figure 37. Classification of the developed PHM models.

shows in graphical form the models that have been developed so far within the PHM project. These models have been classified into three categories as follows:

- *Component models*: designed to model component reliability/availability and costs for each component

- *System models*: designed to model system reliability/availability (i.e., PRA and GRA models) and costs; the `MCSsolver` (see Section 7) which performs the actual probability calculation belongs also to this class
- *Architecture models*: designed to link the component and system models together depending on the application of interest (i.e., the logic and ensemble models).

Lastly, we investigate how data can be enhanced by employing simulator tools. We follow the philosophy that an artificial intelligence bounded only by data is doomed to fail. Learning is coming not only from data but especially from analytical models (e.g., system codes). This learning can easily compensate for lack of, erroneous, or imprecise data and thus provide more robust predictive capabilities.

In FY-21 we are planning to deploy what has been developed during FY-19 and FY-20 in the RIAM and PHM projects for specific use cases that are of interest to our industry partners. We will focus even more on data assimilation and integration with our developed models and methods.

REFERENCES

[1] E. Zio, M. Compare, "A Snapshot on Maintenance Modeling and Applications," *Marine Technology and Engineering*, **2**, pp. 1413-1425 (2013).

[2] Nuclear Energy Institute (NEI), "Critical Component Reduction," NEI Efficiency Bulletin EB 16-25, Washington, DC (2016).

[3] Institute of Nuclear Power Operations, "AP-913 – Equipment Reliability Process Description, Revision 6," Atlanta, GA (2018) (limited distribution).

[4] Value Based Maintenance; NEI Efficiency Bulletin EB 17-03a; 26 January 2017; Nuclear Energy Institute; Washington, DC

[5] Nuclear Energy Institute (NEI), "Risk-Informed Technical Specifications Initiative 5b, Risk-Informed Method for Control of Surveillance Frequencies," NEI Report 04-10, Revision 1, Washington, DC (2007)

[6] D. Mandelli, Z. Ma, R. Youngblood, S. St Germain, C. Smith, P. Talbot, S. Hess, D. Dube, C. Pope, J. Miller, M. Robbins, D. Das, M. Azarian, and J. Coble; Light Water Reactor Sustainability Program: Plant Integral Risk-informed System Health Program; INL-EXT-19-55808; Idaho National Laboratory; Idaho Falls, ID

[7] Nuclear Energy Institute (NEI), "Industry Guideline for Monitoring the Effectiveness of Maintenance at Nuclear Power Plants, Revision 4b", NEI Report NEI 93-01, Washington, DC (2015)

[8] *A Brief History of Equipment Reliability Improvement Efforts in the U.S. Nuclear Electric Generating Industry*, EPRI, Palo Alto, CA: 2019. 3002012920

[9] PMBD v 3.1 (Preventative Maintenance Basis Database, PMBD.epri.com); EPRI, Palo Alto, CA: 2018. 3002005428

[10] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J, Tu, "A Framework for Validation of Computer Models," *Technometrics*, **49**, no.2, pp. 138-154 (2007).

[11] Z. Chen, Y. Liu, and H. Sun, "Deep Learning of Physical Laws from Scarce Data," (2020) [https://arxiv.org/abs/2005.03448].

[12] R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*, CRC Press (2020).

[13] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **63**, no. 3, pp. 425-464 (2001).

[14] C. Montalvo, I. Pazsit, H. Nylen, and V. Dykin, "First evidence of the pivotal motion (tilting mode) of the core barrel in the Ringhals-4 PWR," in *proceedings of PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*, volume 4 (2016).

[15] I. Pazsit, J. Karlsson, and N. S. Garis, "Some Developments in Core-Barrel Vibration Diagnostics," *Annals of Nuclear Energy*, **25**, no. 13, pp. 1079-1093 (1998).

[16] I. Pazsit, H. Nylen, and C. Montalvo Martin, "Refined method for surveillance and diagnostics of the core barrel vibrations of the Ringhals PWRs," in *proceedings of PHYSOR 2014 - The Role of Reactor Physics Toward a Sustainable Future* (2014).

[17] G. Por, "Reactor noise analysis applications and systems in WWER-440 and WWER-1000 type PWRs," Lecture material for IAEA training course in Slovakia (1998).

[18] M. Raissi, P. Perdikaris, and G Karniadakis, "Inferring Solutions of Differential Equations Using Noisy Multi-Fidelity Data," *Journal of Computational Physics*, **335**, pp. 736-746 (2017).

[19] M. Raissi, P. Perdikaris, and G. Karniadakis, "Machine Learning of Linear Differential Equations Using Gaussian Processes," *Journal of Computational Physics*, **348**, no. 1, pp. 683-693 (2017).

[20] A. Trenty, "Operational Feedback on Internal Structure Vibration In 54 French PWRs During 300 Fuel Cycles," *Progress in Nuclear Energy*, **29**, no. 3-4, pp. 347-356 (1995).

[21] A. Ramos, J. Ferreira, J. Barcelo, "Model-Based Systems Engineering: An Emerging Approach for Modern Systems," *IEEE Transactions on Systems Man and Cybernetics*, **42**, no. 1, pp. 101-111 (2011).

[22] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliati, C. Wang, P. W. Talbot, D. P. Maljovec, C. Smith, "RAVEN User Guide," INL Technical report INL/EXT-18-44465 (2020). [https://raven.inl.gov/SitePages/Overview.aspx]

[23] J. Lee and N. J. McCormick, *Risk and Safety Analysis of Nuclear Systems*, Wiley edition (2011).

[24] W. Yan, L. Yu, "On Accurate and Reliable Anomaly Detection for Gas Turbine Combustors: A Deep Learning Approach," *Proceedings of Annual Conference of the Prognostics and Health Management Society*, Coronado, CA, USA (2015).

[25] M. Zhao, M. Kang, B. Tang, M. Pecht, "Deep Residual Networks with Dynamically Weighted Wavelet Coefficients for Fault Diagnosis of Planetary Gearboxes," *IEEE Transactions on Industrial Electronics*, **65**, no. 5, pp. 4290–4300 (2018).

[26] H. Shao, H. Jiang, H. Zhang, T. Liang, "Electric Locomotive Bearing Fault Diagnosis Using a Novel Convolutional Deep Belief Network," *IEEE Transactions on Industrial Electronics*, **65**, no. 3, pp. 2727–2736 (2018).

[27] Z. Liu, Z. Jia, C.-M. Vong, S. Bu, J. Han, X. Tang, "Capturing High-Discriminative Fault Features for Electronics-Rich Analog System Via Deep Learning," *IEEE Transactions on Industrial Informatics*, **13**, no. 3, pp. 1213–1226 (2017).

[28] J. Tian, C. Morillo, M. H. Azarian, M. Pecht, "Motor Bearing Fault Detection Using Spectral Kurtosis-Based Feature Extraction Coupled with K-Nearest Neighbor Distance Analysis," *IEEE Transactions on Industrial Electronics*, **63**, no. 3, pp. 1793–1803 (2016).

[29] M. Kang, G. Krishnan Ramaswami, M. Hodkiewicz, E. Cripps, J.-M. Kim, M. Pecht, "A Sequential *K*-Nearest Neighbor Classification Approach for Data-Driven Fault Diagnosis Using Distance- and Density-Based Affinity Measures," in *Data Mining and Big Data*, Springer (2016).

[30] M. Kang, J. Kim, J. -M. Kim, A. C. C. Tan, E. Y. Kim, B.-K. Choi, "Reliable Fault Diagnosis for Low-Speed Bearings Using Individually Trained Support Vector Machines with Kernel Discriminative Feature Analysis," *IEEE Transactions on Power Electronics*, **30**, no. 5, pp. 2786–2797 (2015).

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer ed. (2009).

[32] I. H. Witten, and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 3rd Edition (2011)

[33] M.E. Orchard, G. Vachtsevanos, "A Particle Filtering Framework for Failure Prognosis," in *Proceedings of the WTC2005, World Tribology Congress III*, Washington, DC (2005).

APPENDIX A: NERC-GADS PC-GAR MT PROGRAM

The North American Electric Reliability Corporation (NERC) began a data collection system in 1982. The Generating Availability Data System (GADS) collects data from over 5,000 electric generating units in North America. GADS collects performance information from generating units that have an electrical capacity of at least 20 MW.

The participating utilities follow instructions to report data to NERC, which is then incorporated into GADS. The pc-GAR programs expand the effectiveness of GADS through a specialized data retrieval system. Users of pc-GAR may access anonymous operational and reliability information from more than 1,189 active generating units. The pc-GAR software allows users to conduct data searches with specified criteria of unit design, fuel characteristics, operating data, etc. The software is updated annually and contains data dating back to 1982.

The MT version of pc-GAR contains information for mean-time-between-failure and mean-time-to-repair data [9]. The data obtained from the pc-GAR software are assigned to basic events in the GRA model. The basic events correspond to certain cause codes as defined by NERC. The cause codes are four-digit numbers identifying specific pieces of equipment or trains of equipment (e.g., cause code 4030 corresponds to high pressure steam turbine rotor shaft). The cause codes used in the GRA model include all the NPP critical components reported to GADS. Table 5 shows the cause code ranges for the sub-systems in the nuclear reactor, balance of plant, steam turbine, and generator main systems.

Table 5. Model systems and subsystems.

Nuclear Reactor		Balance of Plant	
Description	Cause Code Range	Description	Cause Code Range
Core/Fuel	2010-2090	Condensing System	3110-3199
Control Rods and Drives	2110-2160	Circulating Water Systems	3210-3299
Reactor Vessel and Internals	2170-2199	Condensate System	3310-3399
Reactor Coolant System	2200-2399	Feedwater System	3401-3499
Steam Generators and Steam System	2400-2599	Heater Drain Systems	3501-3509
Core Cooling/Safety Injection	2600-2649	Extraction Steam	3520-3529
Electrical Safety Systems	2650-2699	Electrical	3600-3689
Containment System	2700-2799	Auxiliary Systems	3810-3899
Chemical and Volume Control/Reactor Water Cleanup	2805-2819	Miscellaneous (Balance of Plant)	3950-3999
Nuclear Cooling Water Systems	2820-2839	Steam Turbine	
Auxiliary Systems	2840-2890	Description	Cause Code Range
Miscellaneous (Reactor)	2900-2999	High Pressure Turbine	4000-4099
Generator		Intermediate Pressure Turbine	4100-4199
Description	Cause Code Range	Low Pressure Turbine	4200-4250
Generator	4500-4580	Valves	4260-4269
Exciter	4600-4609	Piping	4270-4279
Cooling System	4610-4650	Lube Oil	4280-4289
Controls	4700-4750	Controls	4290-4309
Miscellaneous (Generator)	4800-4899	Miscellaneous (Steam Turbine)	4400-4499

Some cause codes were intentionally left out of VERT. Any cause code specifically related to CANDU reactors was not used. Cause codes specifically related to BWR units were not included in the PWR version and vice-versa. If a cause code was not reported for all nuclear units from 1985-2019 the code was not used.

Several cause codes relating to the “Electrical” sub-system were omitted due to the vast variations. The cause codes had different variations depending on the voltage of the components. The voltages where variant anywhere from 480 volts to 12 kilovolts.

APPENDIX B: VERT USER'S MANUAL

The goal of the following process is to produce a .xml file, which will then be used as an input file for the RAVEN program. RAVEN will then iterate through SAPHIRE and generate information regarding the specific plant in graph and excel sheet forms. Aside from the SAPHIRE and RAVEN programs, initially everything will be contained in two ".zip" files. One ".zip" file, "MAC.zip", will need to be placed in the RAVEN directory as shown. The other ".zip" file, "DEMO.zip" will need to be placed in the SAPHIRE 8 directory as shown. These files contain all the documentation needed following installation of SAPHIRE and RAVEN. Figure 38 displays the contents of the "DEMO.zip" file after it has been extracted. The demo file needs to be in the "raven" directory or the program will not run.

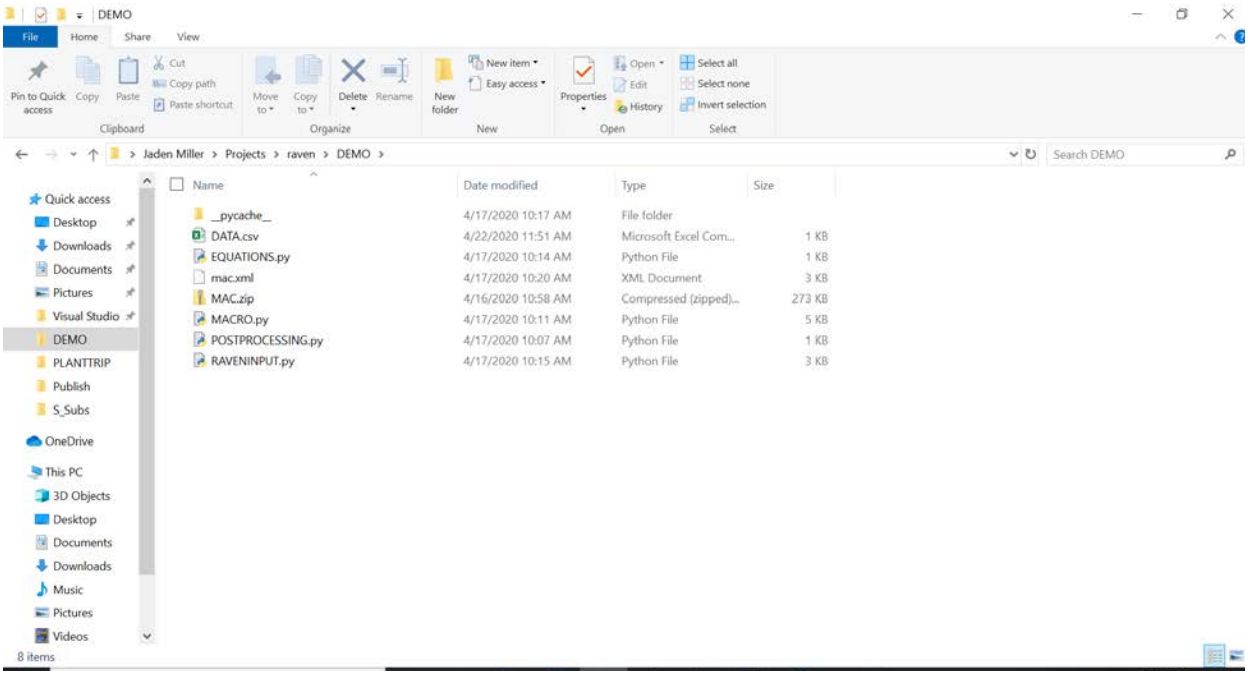
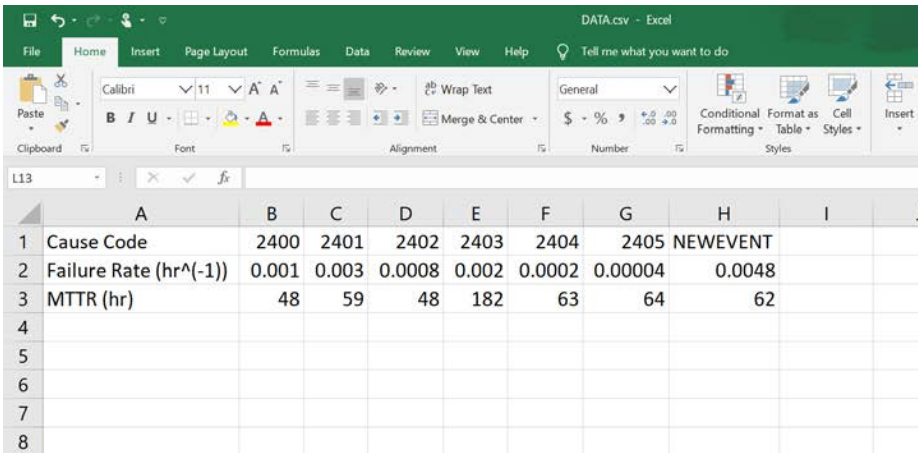


Figure 38. Contents of the VERT documentation.

"DATA.csv" should be opened in excel or another spreadsheet editing software. "DATA.csv" contains the data associated with all the basic events contained in the model. To insert additional basic events, follow the formatting of the spreadsheet and type in the appropriate data. The "Cause Code" identified in the spreadsheet needs to be included in the basic event name in the SAPHIRE model. An example of inserting a new event into SAPHIRE is shown later. Figure 39 shows the "DATA.csv" spreadsheet with a new event typed in.



	A	B	C	D	E	F	G	H	I
1	Cause Code	2400	2401	2402	2403	2404	2405	NEWEVENT	
2	Failure Rate (hr^(-1))	0.001	0.003	0.0008	0.002	0.0002	0.00004	0.0048	
3	MTTR (hr)	48	59	48	182	63	64	62	
4									
5									
6									
7									
8									

Figure 39. Input data spreadsheet for the program.

After "DATA.csv" has been opened and modified as desired for the program, the SAPHIRE project should be opened and run. The SAPHIRE project is contained in the "MAC.zip" file. Before the SAPHIRE project is run the "MAC.zip" file should be copied and relocated to the "SAPHIRE 8" directory on the computer. The contents of the "MAC.zip" file should be extracted in the "SAPHIRE 8" location. Figure 40 shows how the demo project should be placed on the computer.

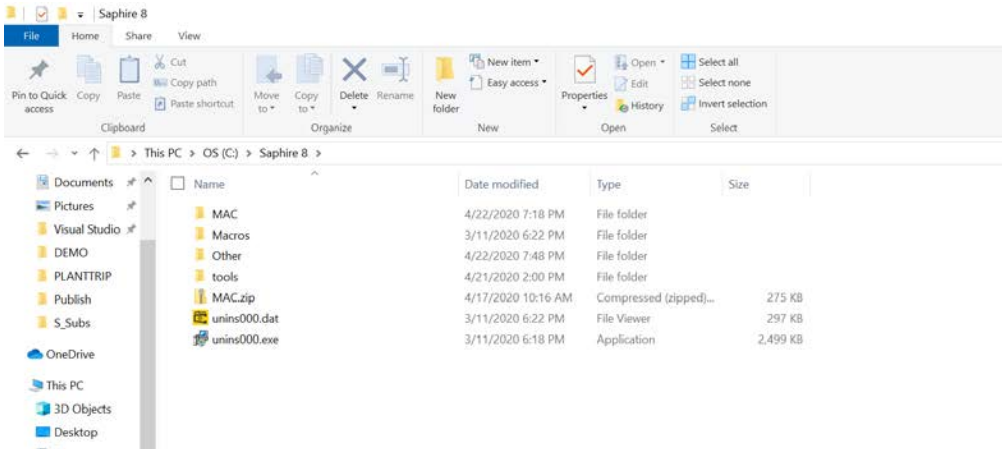


Figure 40. SAPHIRE 8 location for the demo project.

The project should then be opened using SAPHIRE. To do this, first open SAPHIRE. Next, open the demo project by first clicking "File", then "Open existing project…". Then locate the MAC project. Double click the .SRA file (see Figure 41) The project can now be edited as desired.

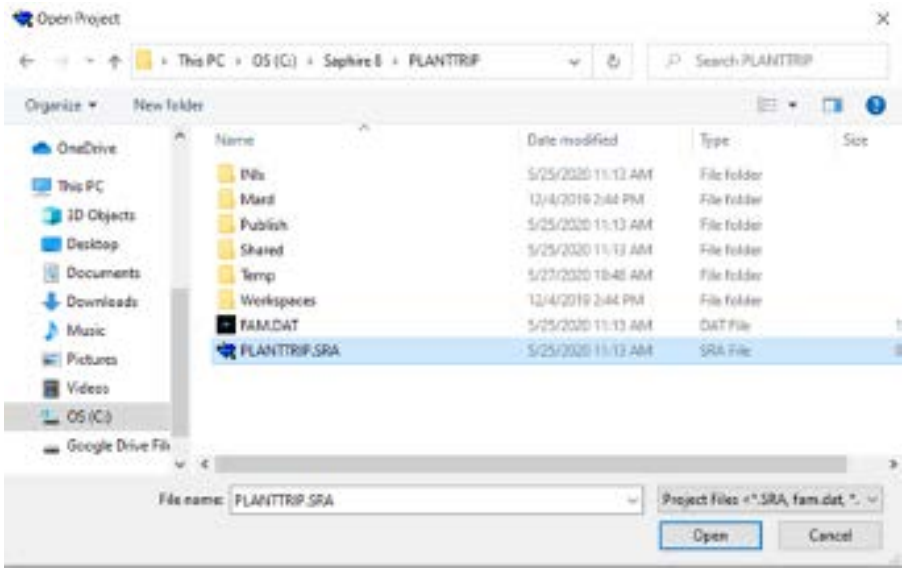


Figure 41. Project .SRA file.

After the project has been edited to satisfy the new events placed in "DATA.csv", a basic event report should be created. Make sure the MTTR value of new basic events is created if different from existing MTTR values. The notation of the MTTR event should be "REVENTNAME"; e.g. for the case of a new event named "NEW", the MTTR event should be "RNEW". If events were not added to "DATA.csv", no new events need to be edited or added in the SAPHIRE project. Publish a basic event report by locating the "Basic Events" window in SAPHIRE and changing the basic event types to "All." To do this first click on the first basic event so it is highlighted. After the first basic event is highlighted press Ctrl + A on the computer keyboard. After all of the basic events are highlighted as shown in Figure 42, click on the "Publish" tab and then the "Basic Event Report" tab.

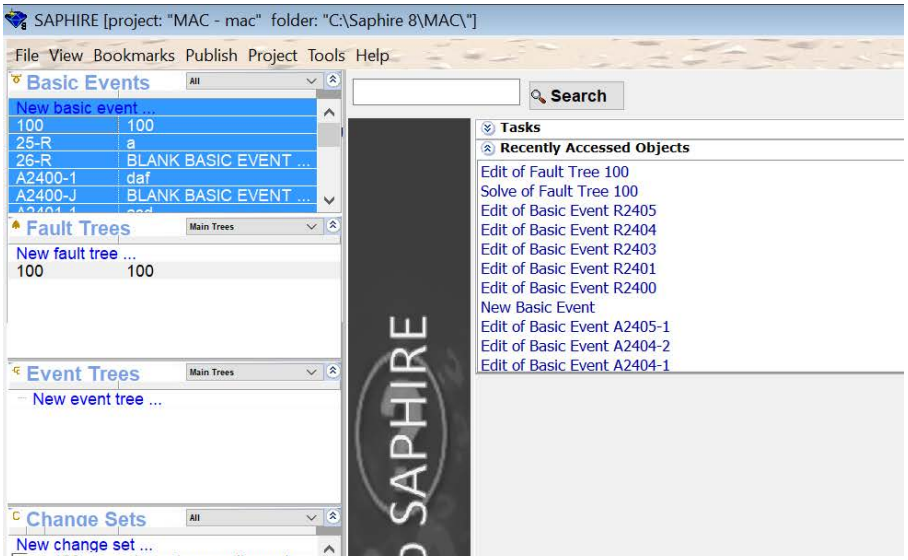


Figure 42. All the basic events highlighted for the basic event report.

The basic event report should be published to an ".xls" spreadsheet and a "Basic Event Listing" report should be selected. The screen should look like Figure 43.

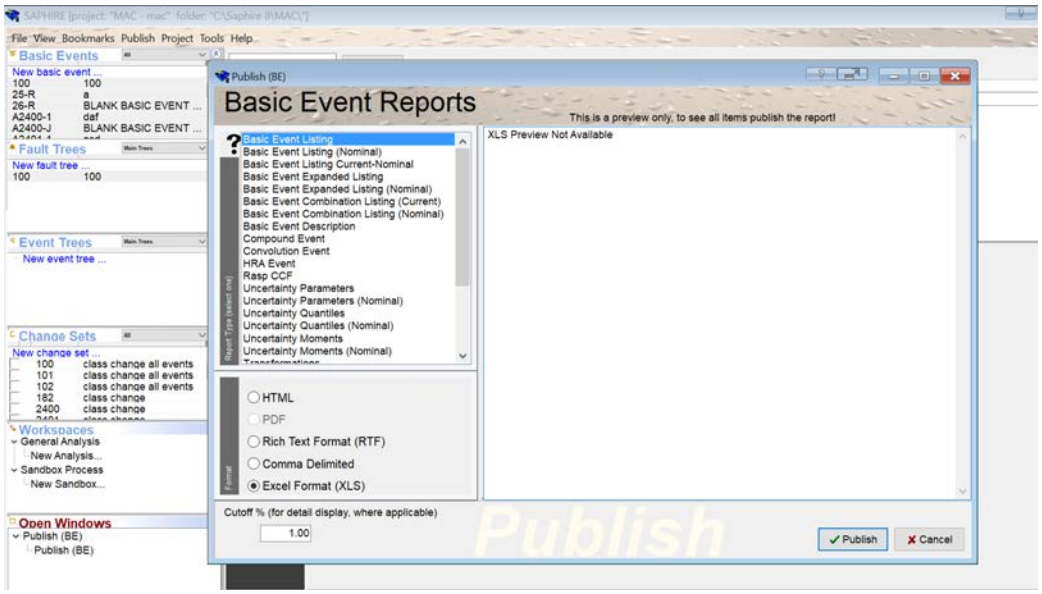


Figure 43. The basic event publishing screen.

The basic event report should be automatically opened in excel or some other spreadsheet editing program. The opened file should then be changed to a ".csv" file and saved as "BE.csv" in the "DEMO" folder. The "DEMO" folder should now contain the "BE.csv" basic event report. Next, locate and run the "VERT" application file. It should be in the "DEMO" folder with the "DATA.csv" and "BE.csv" files. When the VERT application is run it should look like Figure 44. Follow the prompts and run the POSTPROCESSING script. The POSTPROCESSING script should generate a text file named "rules.txt". This file should be created in the "DEMO" folder.

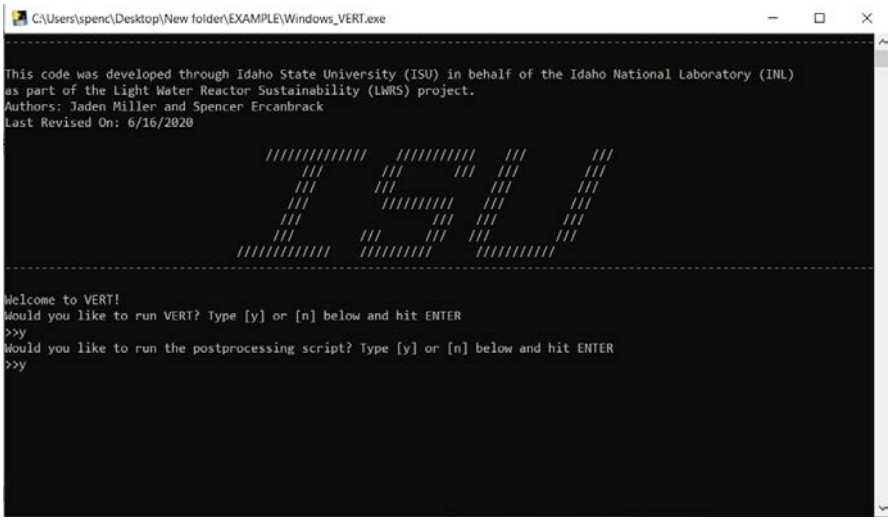


Figure 44. The post processing rules text creating code.

The text inside “rules.txt” needs to be copied into SAPHIRE. Open “rules.txt” with any text editor and click CTRL+A on the keyboard to highlight all the text. The file should look similar to Figure 45. Click CTRL+C simultaneously on the keyboard to copy the text.

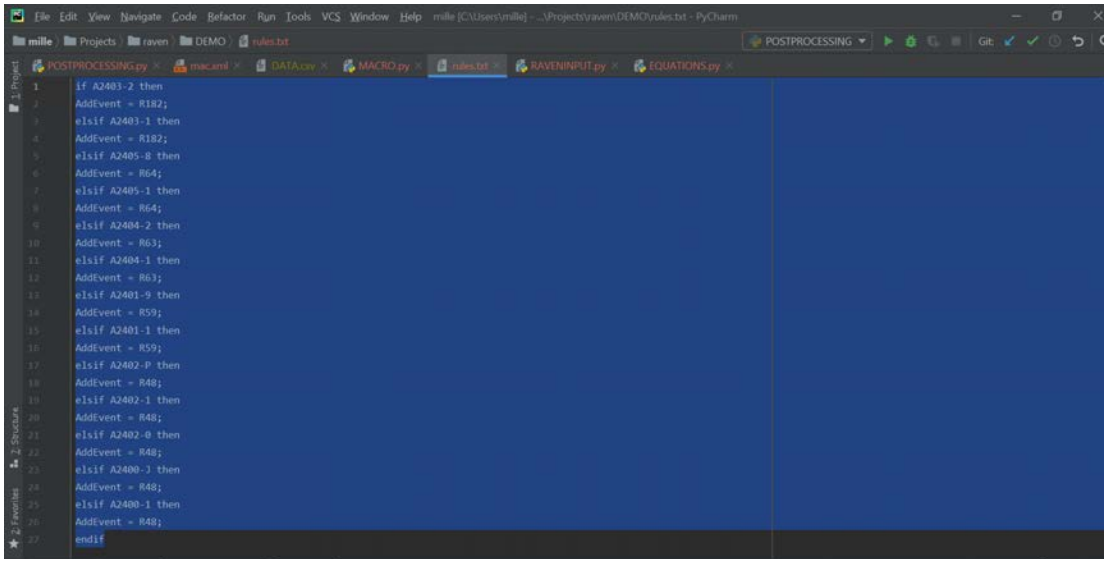


Figure 45. Post processing text file.

The text should be copied to SAPHIRE in the FT post processing rules text editor. To access the post processing rules text editor open the SAPHIRE project and click “Project”, then “Edit Rules”, and then “FT (Post-processing).” Click in the box to select it, then press CTRL+V to paste the post processing text in the box. The text editor should look like Figure 46.

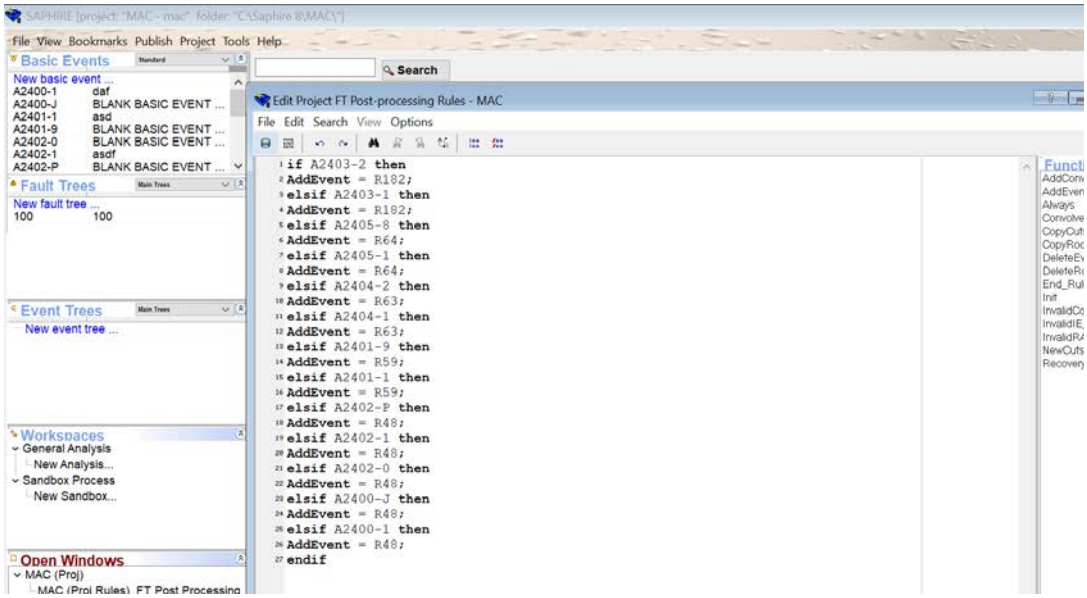


Figure 46. Post processing rules text editor.

Once the text is copied into the post processing text editor, the rules need to be saved and compiled. The “save” and “compile” buttons can be found either in the “File” dropdown menu or on the post processing text editor ribbon located just below the “File” and “Edit” dropdown menus. The post processing text editor can be closed after saving and compiling. The next step is to solve the FTs in two different manners. In the FT window located in SAPHIRE, change the FT type to “All” and highlight every FT by clicking CTRL+A. Right click on the highlighted FTs and click “Solve”. The screen should look like Figure 47.

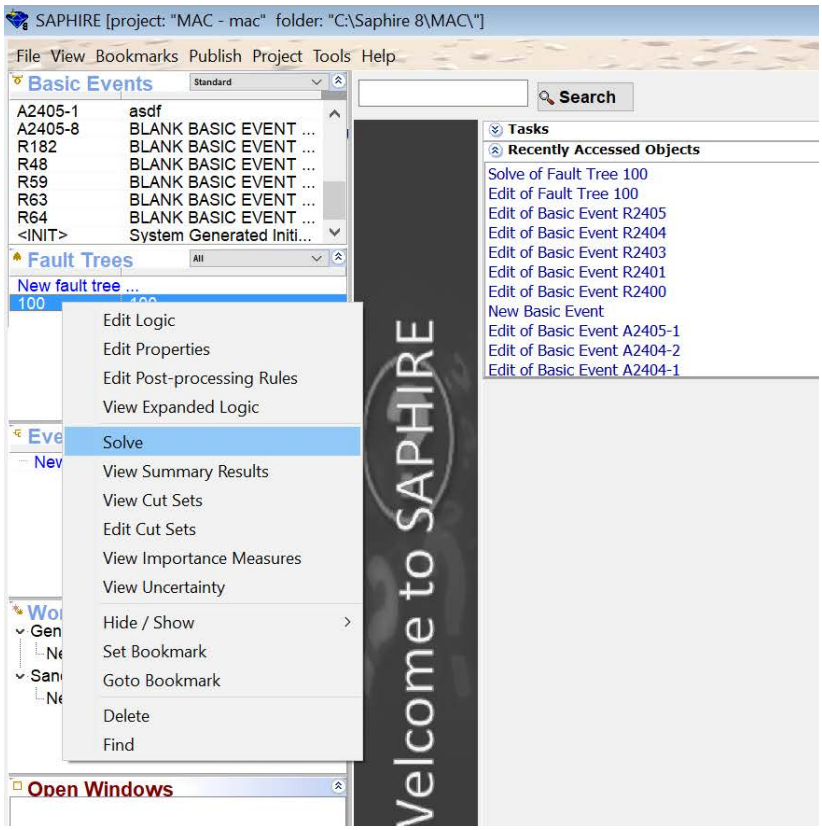


Figure 47. Solving the FTs example.

A window will pop up - make sure to select all the solution steps and ensure the rare event quantification method is selected. The screen should appear the same as Figure 48.

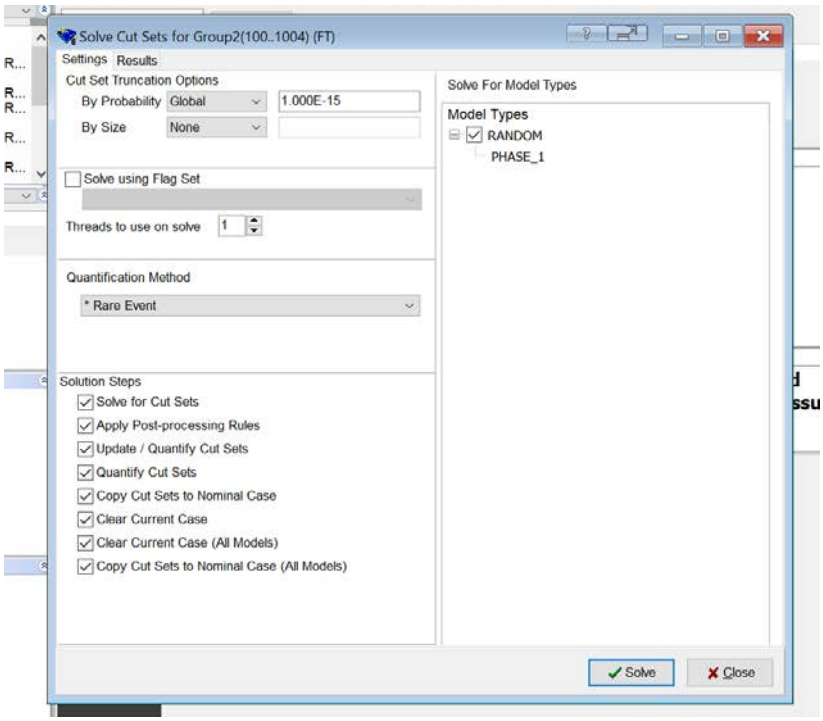


Figure 48. Solution step one.

Depending on the project size, SAPHIRE may take up to twenty minutes to solve for the cut sets. Once the cut sets are solved, the screen should look like Figure 49.

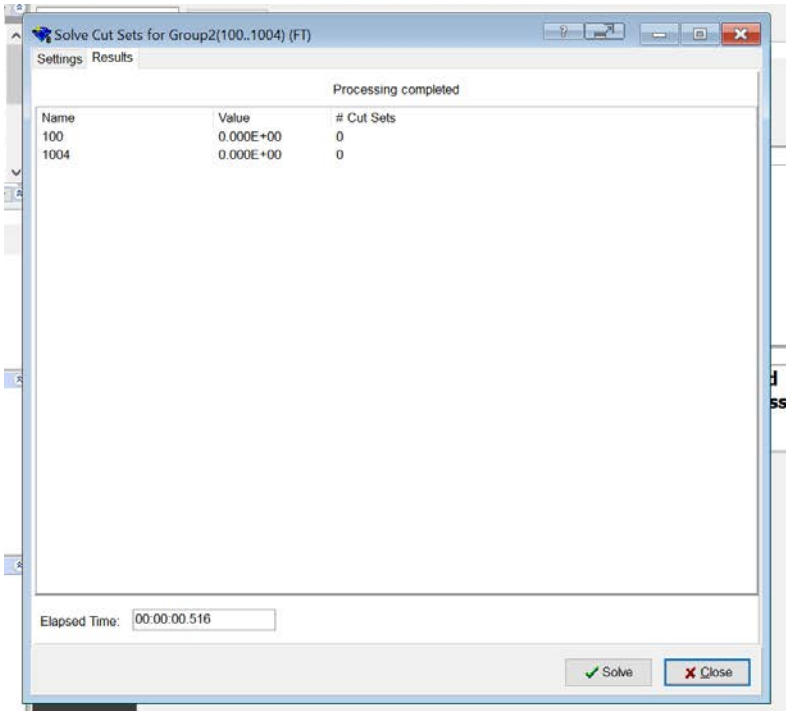


Figure 49. Solved FT s in solution step one.

The window can be closed after the FT are solved. The FT s need to be solved again using different solution steps. The solution steps selected should be the same as shown in Figure 50.

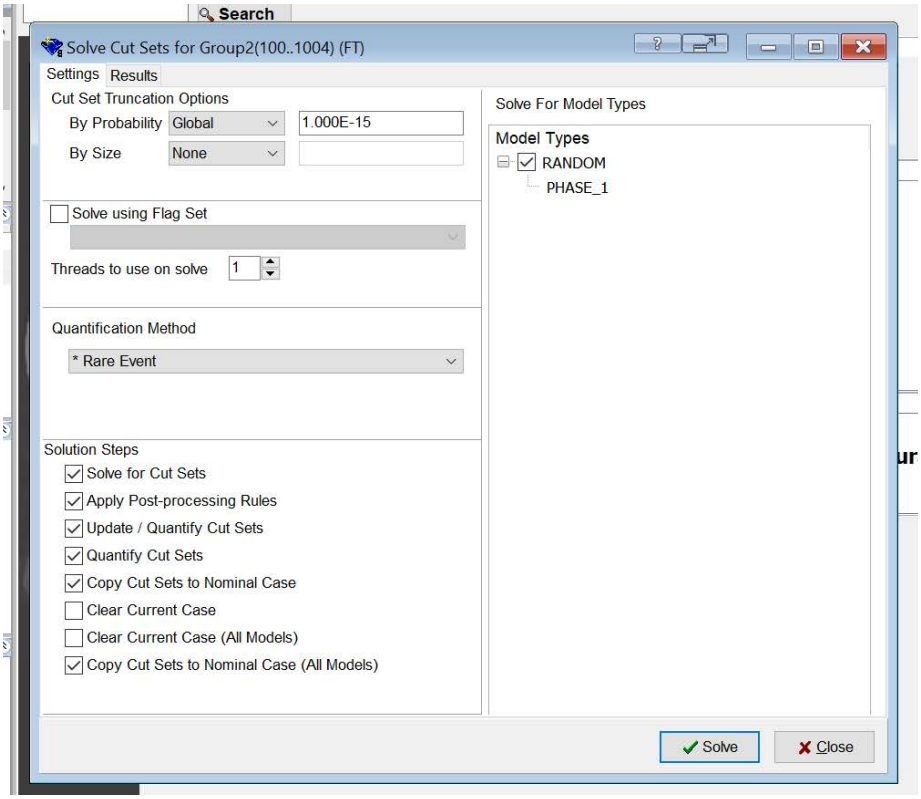


Figure 50. Solution step two.

The same window shown in Figure 49 will appear after the FT s are solved, differing in the results. The window and SAPHIRE can now be closed. After SAPHIRE is closed, the contents in the “MAC” project need to be sent to a compressed “.zip” file. To do this first open the “MAC” project and highlight all the contents by pressing CTRL+A. Then right click on the contents and select “send to compressed (zipped) folder”. A message stating “Windows was unable to add one or more empty directories to the compressed (zipped) folder” will appear. Select, “ok”, and rename the newly created .zip folder “MAC”. Move this .zip folder to the “SAPHIRE 8” directory. If SAPHIRE is open when the contents are sent to a .zip folder, an error message stating, “File not found or no read permission”, will appear. Simply close SAPHIRE and try again. Return to the VERT application and run the MACRO script by following the prompts. The screen should look like Figure 51.

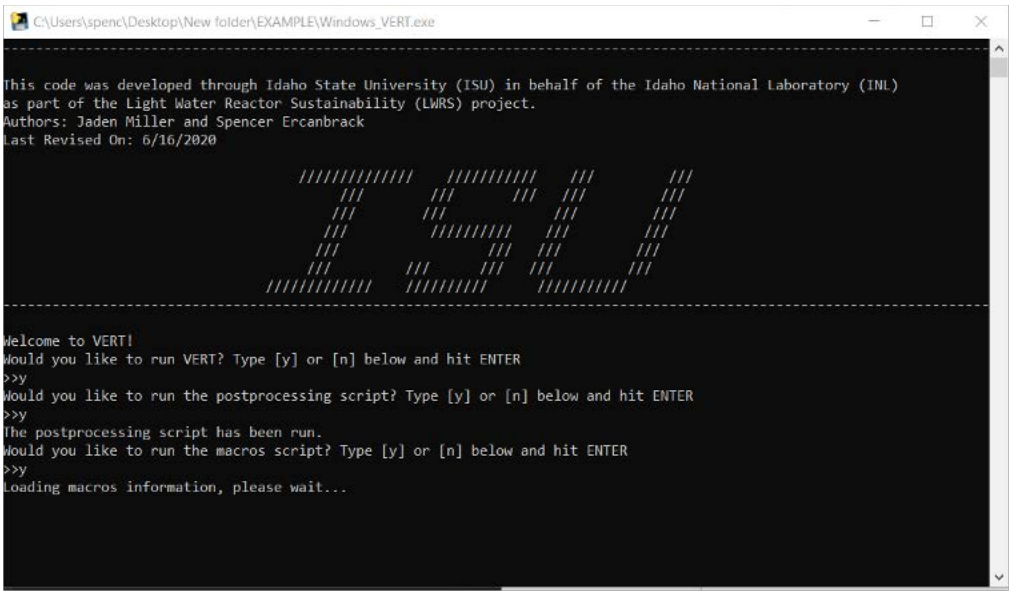


Figure 51. VERT application macros script.

Run the EQUATIONS script by following the prompts in the VERT application. The screen should look like Figure 52.

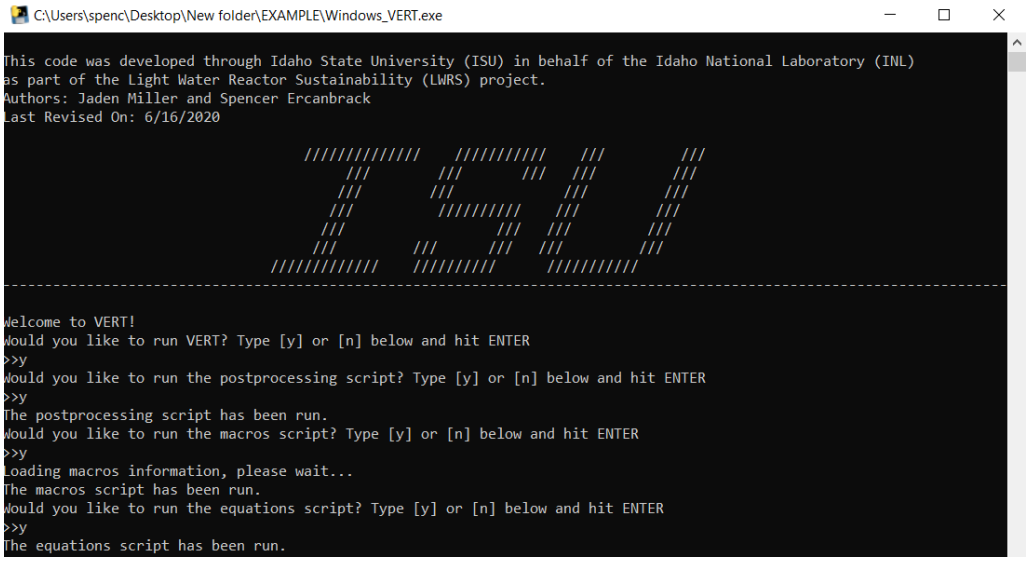


Figure 52. VERT Application equations script.

The final script to run is the RAVENINPUT script, which can be run by following the prompts in the VERT application. The screen should look like Figure 53.

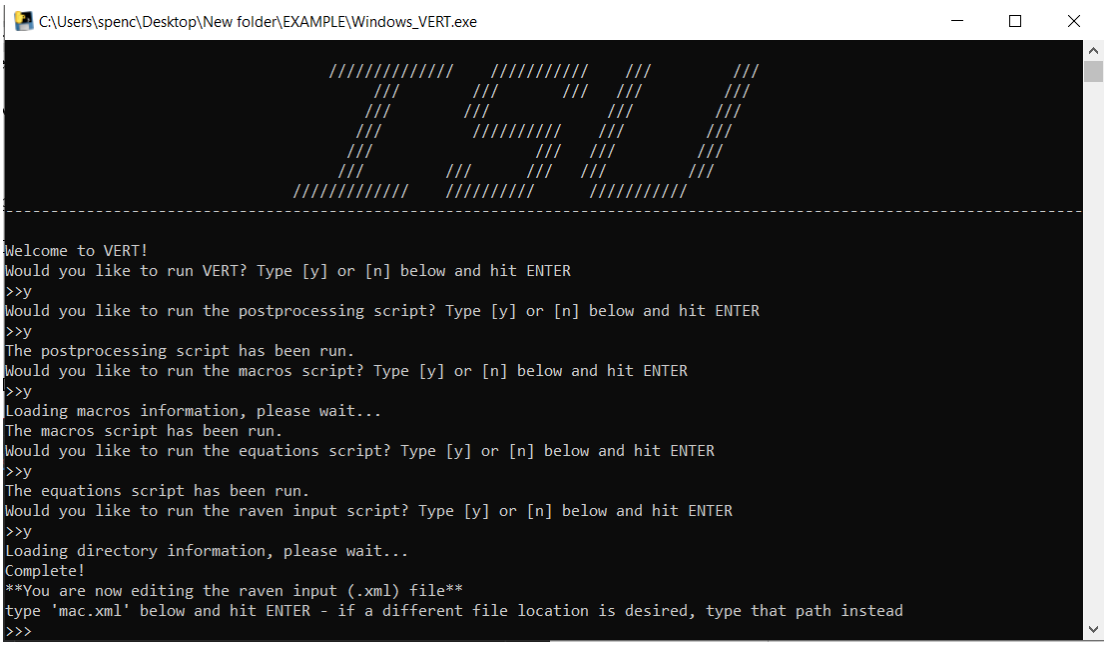


Figure 53. RAVEN xml input file editor code.

After the RAVENINPUT script has run, exit the program by following the prompts. Open the "mac.xml" file and ensure lines 9, 10, and 16 have the correct locations for the macro file, the project ".zip" file, and the SAPHIRE executable, respectively. The program is now ready to run. Open raven and run the "mac.xml" file. Open raven by locating the "raven_framework.bat" file in the "raven" directory and double clicking it. The screen should look like Figure 54 when RAVEN is run.

Figure 54. RAVEN executing the program.

The output from VERT is contained in the user specified working directory. The output files are images or spreadsheets with the results from the program execution. The file containing all of the information is "SaphirePrint.csv." The file contains all of the values for the basic events, risk values, and the FT results. The values from the spreadsheet are automatically exported into the images. The images are comparisons of the different systems and sub-systems on scatter plots.

APPENDIX C: THE VERT PROGRAM

The fundamental purpose of VERT is to estimate the lost money imposed on NPPs from unavailable SSCs due to failures. The program uses PRA methodologies to evaluate the effects failed components have on a system. The effects of concern in the program are any form of derate occurring from equipment failure. FTs are arranged to depict a system and probabilistic tools are used to estimate the likelihood of an NPP derate. The consequences of the events are integrated into the program to quantify the associated risk of the failed components.

VERT utilizes two different software packages developed by the Idaho National Laboratory (INL). The software package used to evaluate the FTs for LWR systems is Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE). SAPHIRE creates and analyzes PRA, primarily for NPPs. The RAVEN package is used to integrate time-dependence and consequences into the program. RAVEN is a versatile framework used for uncertainty quantification, regression analysis, PRA, data analysis, and model optimization. SAPHIRE and RAVEN are coupled together in the program to perform time-dependent economic risk assessments.

SAPHIRE is used to develop fault trees to represent an NPP system layout. Common PRA methodologies are used to build the FTs. PRA is much more complex in nature than the GRA methods used in VERT. PRA analyzes systems based on the ability to perform a safety function successfully. The only function of concern in an economic program is operating at full or reduced power to maximise revenue. Figure 55 may be used to illustrate the difference. The function of the feedwater system in PRA often identified is heat removal. The system may need two of the three pumps, both heaters or a single heater and a bypass loop, and all four steam generator loops to remove an adequate amount of heat.

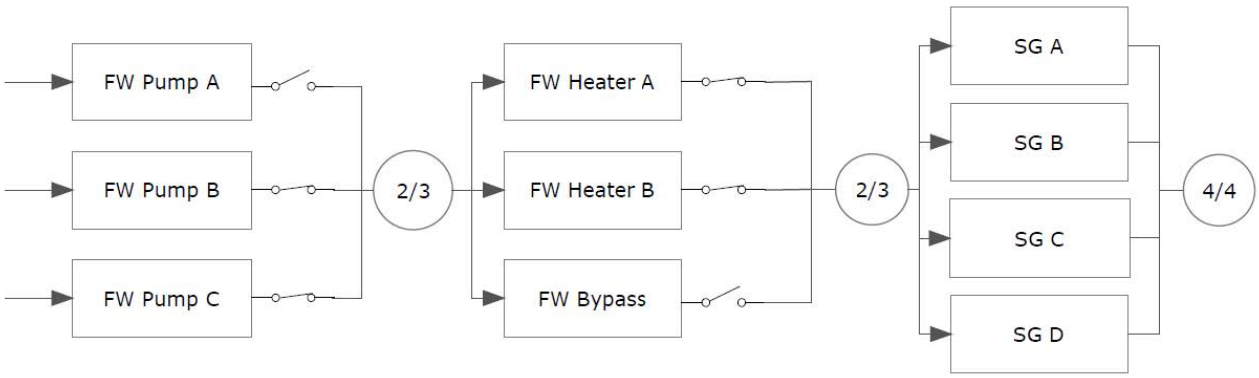


Figure 55. Availability block diagram of a feedwater system for full power.

Failure to maintain that status may cause inadequate heat removal from the system. However, there are supporting systems to perform the function, such as the auxiliary feedwater system. Therefore, both the feedwater system and the auxiliary feedwater system must fail to lose heat removal. The NPP may be tripped or derated upon loss of the feedwater system even if the auxiliary system operates successfully.

These types of events are evaluated in VERT. For example, in Figure 55 the logic is represented as full power of the system. The system may also be evaluated at a fractional output power level as well. To support 50% output power the feedwater system may only need one of the three pumps, one of the heaters or the bypass loop, and two of the four steam generator loops. Fractional levels of NPP electrical capacities are important to evaluate because derates tend to be more common than full plant trips. VERT eliminates the need for multiple models representing the different fractional levels of output capacity. Rather than

having different models created for derate levels of 50%, 30%, 100%, etc, the derate percentage is accounted for in the post-processing of the FT cut sets. The SAPHIRE post-processing tool introduces the consequences into Equation (C1):

$$Risk = Likelihood \times Consequences \quad \text{(C1)}$$

There are many derates that can occur in an NPP and evaluating each possible derate would be time consuming. VERT can evaluate all possible derates using one FT model. Each component included in the model is associated with a derate percentage upon failure of the component. The derate percentage is contained in the spreadsheet included with VERT.

The "Likelihood" value in Equation is obtained through solving the FT model in SAPHIRE. The cut sets' likelihoods are quantified using standard probabilistic methods. The data obtained from pc-GAR give failure frequencies for the basic events in the model. The method for obtaining data from pc-GAR is explained in the next paragraph. The failure probabilities are then determined from Equation(C2). The equation determines the probability of failure without repair for some failure frequency (λ) and some mission time (t).

$$\mathrm{P} = 1 - e^{-\lambda t} \quad \text{(C2)}$$

Failure without repair probabilities were used because derating of the NPP occurs following a critical component failure. Failure with repair probabilities would be used to evaluate models where two or more critical components failure causes the top event to occur.

The data obtained from pc-GAR includes the failure rates, Mean Times To Repair (MTTRs), and the average derates due to equipment failure. When pc-GAR is given search criteria, the results are shown in Table 6 for some arbitrary events.

Table 6. Example output from pc-GAR search.

Event	Cause Code	Time to Repair	Derate
1	3401	36 hr	34%
2	4309	5 hr	100%
3	2010	80 hr	10%
4	2010	41 hr	13%

The total unit service hours are also known for the pc-GAR search criteria. The failure rate for each cause code is determined by finding the total occurrences of each cause code event and dividing by the total unit service hours. The methodology for deriving the failure rate from pc-GAR was adopted from EPRI (Electric Power Research Institute). The MTTR values and average derates for each cause code is simply the arithmetic average of all the occurrences.

There are multiple derate scenarios that components and trains are subject to. The scenarios with credible risk are included in the model. Figure 56 can be used as an example of excluded scenarios. If both

feedwater pump trains fail, assuming each pump supports 50% power, a plant trip would occur. If one of the pump trains fail, a 50% derate would occur.

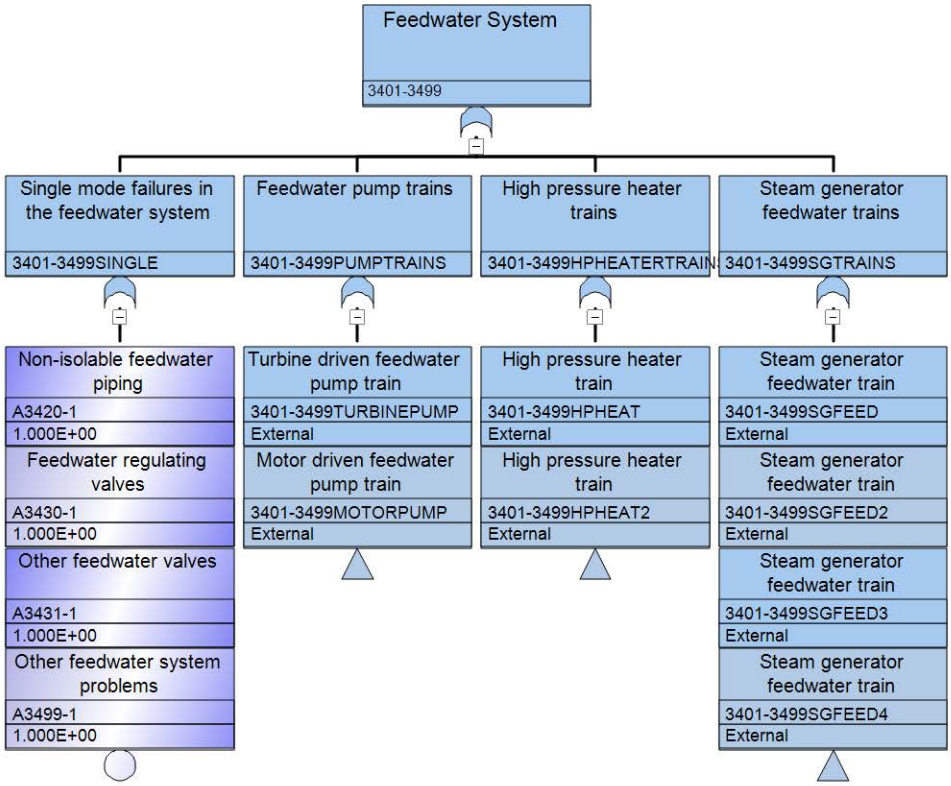


Figure 56. Feedwater system FT logic.

FT logic contributing to little risk was eliminated from the model. The failure of two or more operating trains at the same time does not significantly affect risk. Returning to the feedwater pumps as an example, the economic risk due to both feedwater pump trains failing is negligible compared to the risk imposed from each individual train. Table 7 shows the calculations comparing the likelihood, consequences, and risk from failures of the feedwater pump drives.

Table 7. Risk calculations for feedwater pump drive scenarios.

Component	λ (hr^{-1})	MTTR (hr)	Average Derate
Pump A Drive (motor driven)	6.70E-06	45.4	50%
Pump B Drive (turbine driven)	2.22E-05	40	50%
Scenario	**Probability**	**Consequences ($)**	**Risk ($/yr)**
Pump A Drive Fails	5.70E-02	7.66E+05	4.37E+04
Pump B Drive Fails	1.77E-01	6.75E+05	1.19E+05
Pump A&B Drives Fail	2.70E-07	4.89E+06	1.32E+00

Although the consequences of both drives failing simultaneously are higher by about an order of magnitude, the likelihood is so low the risk becomes negligible. The likelihood is much lower because the failure probabilities of the feedwater pump drives now have time to be repaired. If Pump A or Pump B fail independently, a 50% derate will occur without time to repair the components. However, for the 100% derate to occur, both trains must be unavailable simultaneously. This introduces the time to repair a failed component before the other fails. In these situations, the failure probability is expressed as Equation (C3) :

$$P = \frac{\lambda\tau}{1+\lambda\tau}\left(1 - e^{-\left(\lambda+\frac{1}{\tau}\right)t}\right) \tag{C3}$$

The mission time is t, the failure rate is λ, and the mean time to repair is τ. The risk of non-single mode failures with these assumptions is negligible and thus not included VERT.

VERT can integrate degradation into the component analysis. VERT is currently capable of supporting two different equation types: linear and exponential degradation of lambda (failure rates):

$$\lambda(t) = \lambda_0(1 + bt) \tag{C4}$$

$$\lambda(t) = \lambda_0 e^{bt} \tag{C5}$$

The equation type and degradation constant, b, is specified by the user in the data spreadsheet.

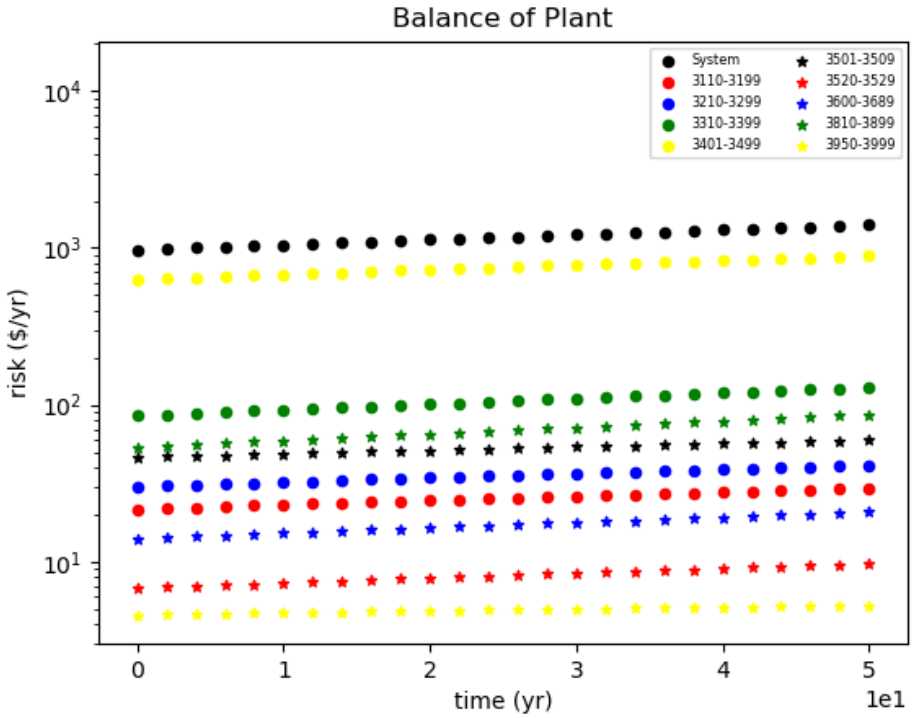


Figure 57. Balance of plant system risk.

The output from VERT is contained in the user specified working directory. The output files are images or spreadsheets with the results from the program execution. The file containing all of the information is "SaphirePrint.csv." The file contains all of the values for the basic events, risk values, and the FT results. The values from the spreadsheet are automatically exported into the images. The images are comparisons of the different systems and sub-systems on scatter plots. The Balance of Plant system is shown as an example in Figure 57.

APPENDIX D: BASIC SAPHIRE USER'S GUIDE

SAPHIRE is a program developed by INL which can perform PRA functions. To accomplish the work detailed in this report, this user's guide will focus on the basics of using SAPHIRE to modify FT logic and generate the associated reports.

Opening SAPHIRE

The first step to using SAPHIRE is to open the program. This can be done by double clicking the application logo or by locating and running the SAPHIRE.exe file. The opening screen should look like Figure 58.

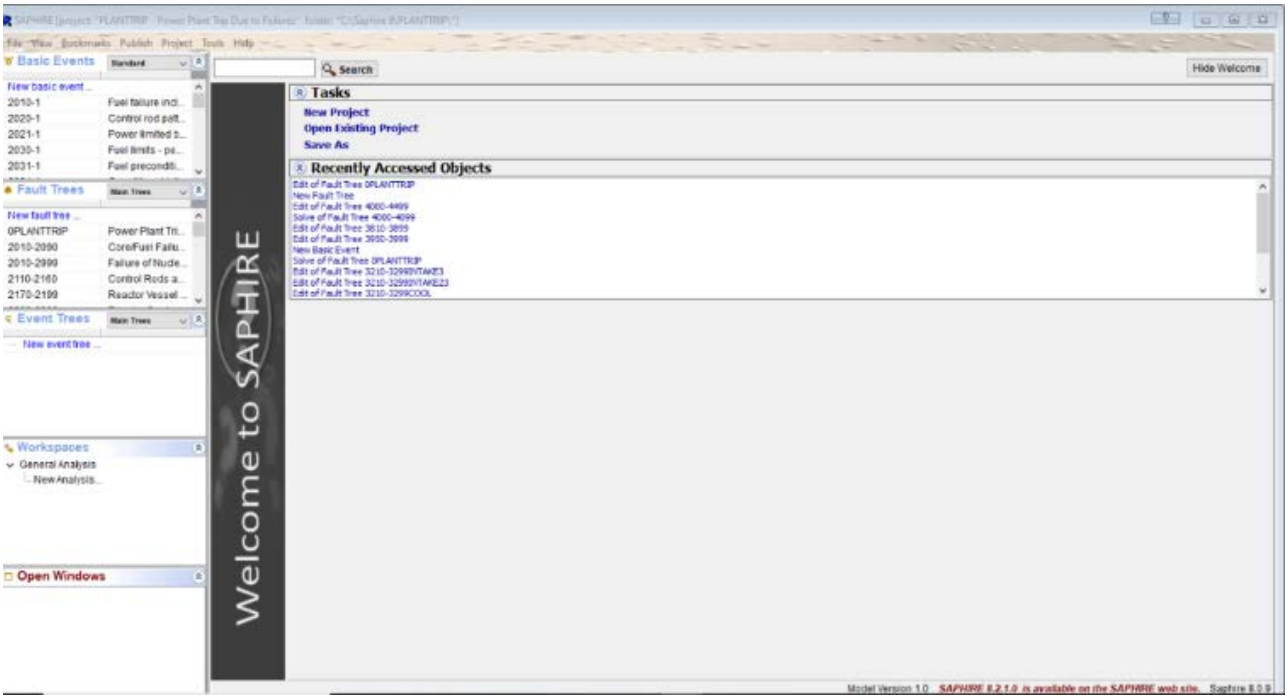


Figure 58. SAPHIRE welcome screen.

Locating/Opening a Project

The next step is to open a new or existing project. A new project can be opened by selecting the "open new project" option on the main screen. An existing project can be opened by first selecting the "File" dropdown menu in the top-left corner of the main menu and then selecting the "Open existing project…" option. Locate the project folder, enter the folder, and select the project's .sra file. See Figure 59.

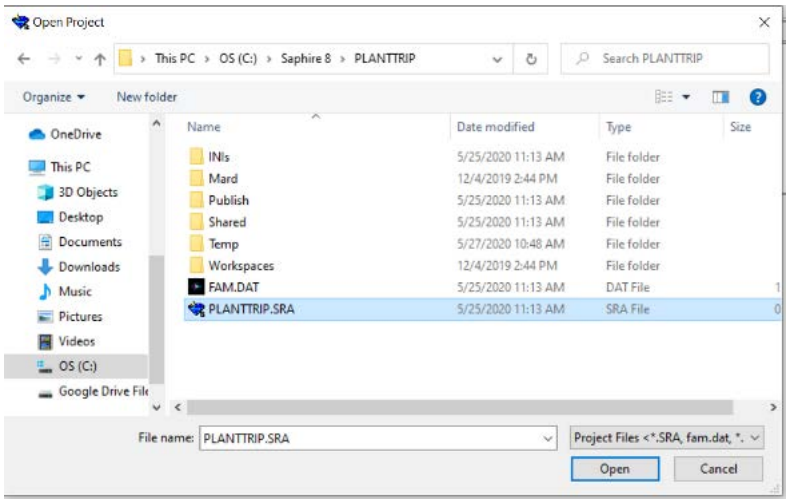


Figure 59. Example .sra file.

This will open the existing project for editing.

Locating FT Section/Selecting A FT

SAPHIRE can be used to create ETs and FTs. This guide will focus on the latter. The FT menu is located on the left side of the main menu, underneath the dropdown menus and the basic event menu. The FT menu displays existing FTs and provides an option to either filter existing FTs or create new FTs.

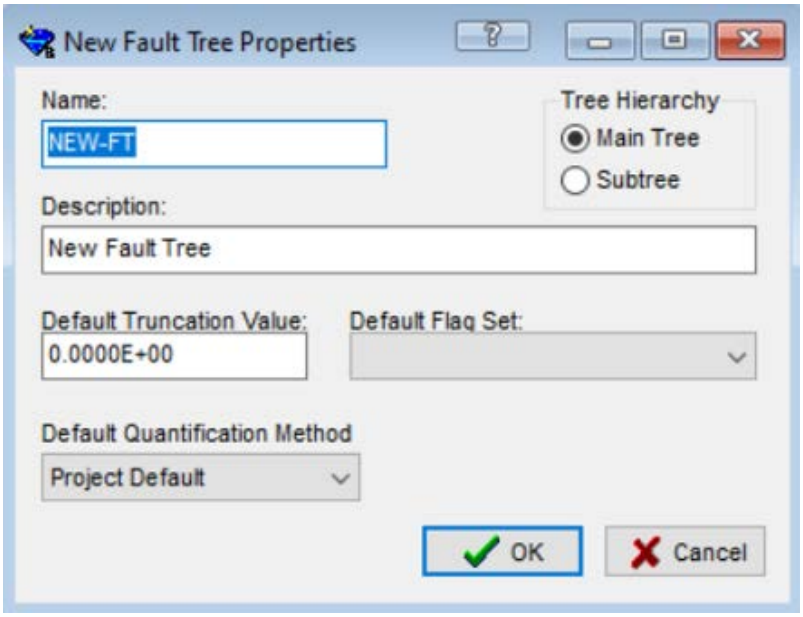


Figure 60. New FT window.

To create a new FT simply double click the “New FT” button at the top of the FT menu. A window similar to Figure 60 will pop up. After the information is input, click “OK”. To look at an existing FT,

simply double click it. When a FT is opened a new window displaying the FT will open. See Figure 61 for an example.

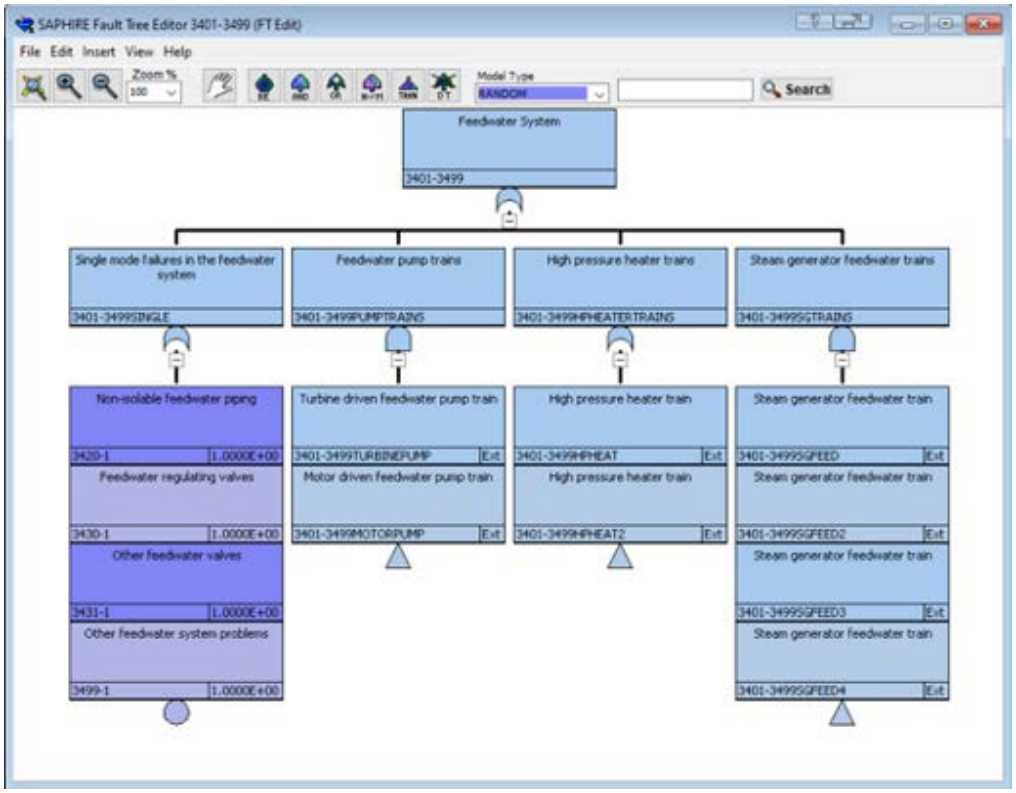


Figure 61. Example FT.

Both gates and events can be easily added, deleted, or edited. To add a gate, first select the type of gate at the top of the FT editing window as shown in Figure 62.

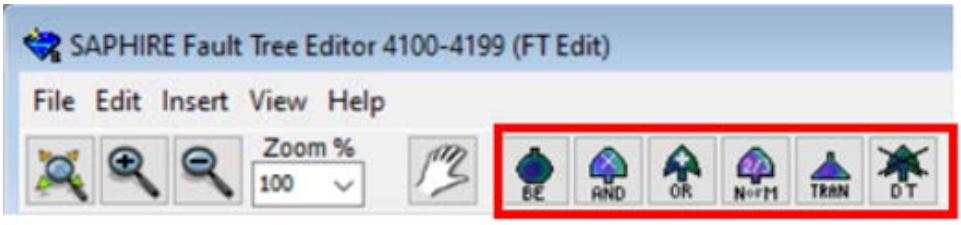


Figure 62. FT gate locations.

Next, place the cursor over a part of the gate shown in Figure 63. The cursor will display a “null” symbol until it is placed over an area where a gate can be added. To add a basic event, select the basic event option (from the same place where the gate options are) and follow the same procedure.

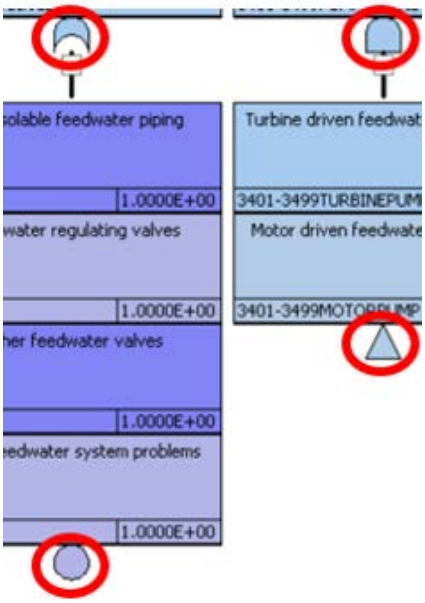


Figure 63. Valid gate addition locations.

Following FT Transfer Gates

Large FTs will generally be made of several smaller fault trees. Because the entire FT cannot fit on one page, these smaller FTs will be attached to the larger FT using "transfer gates". To follow a transfer gate to the attached FT, simply click on the locations shown in Figure 64.

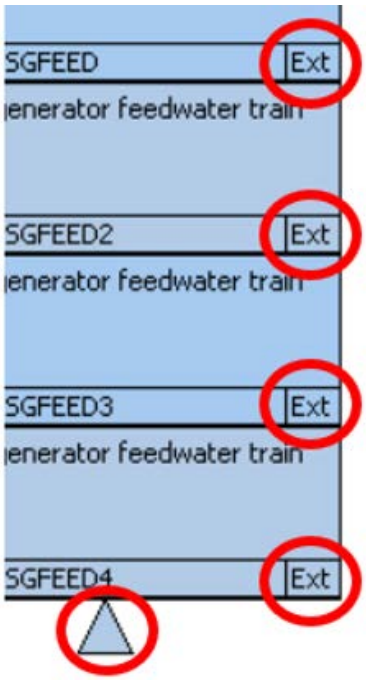


Figure 64. Transfer gate links.

FT Edits

To edit gate or basic event information, simply double click the gate or basic event of interest. This will open a window similar to Figure 65.

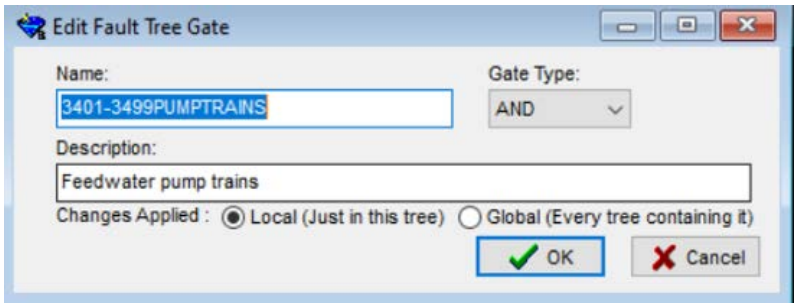


Figure 65. FT edit window.

To delete a gate or basic event, simply click on the gate or basic event of interest and click the "delete" key on the keyboard. Alternatively, right clicking on a gate or basic event will display several options, including "edit" and "delete".

Basic Event Location

The basic event window is located in the top-left side of the SAPHIRE main menu, just below the dropdown menus and just above the FT window. The basic event window functions in the same way as the FT window. It displays all existing basic events, has a dropdown menu containing several filtering options, and has a "New basic event" button. See Figure 66.

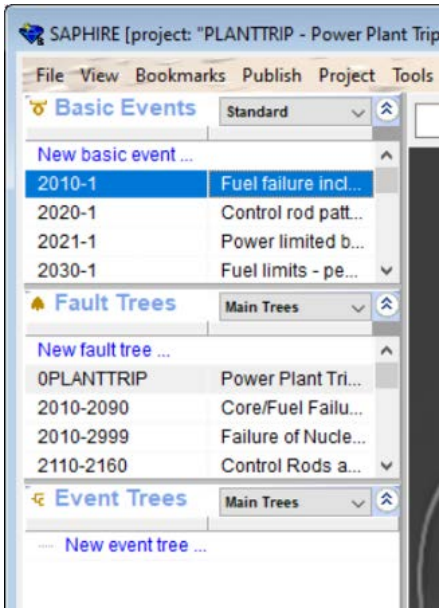


Figure 66. Basic event window location.

Basic Event Edits

When a basic event is opened for editing, a window similar to Figure 67 will open. Each field can be edited.

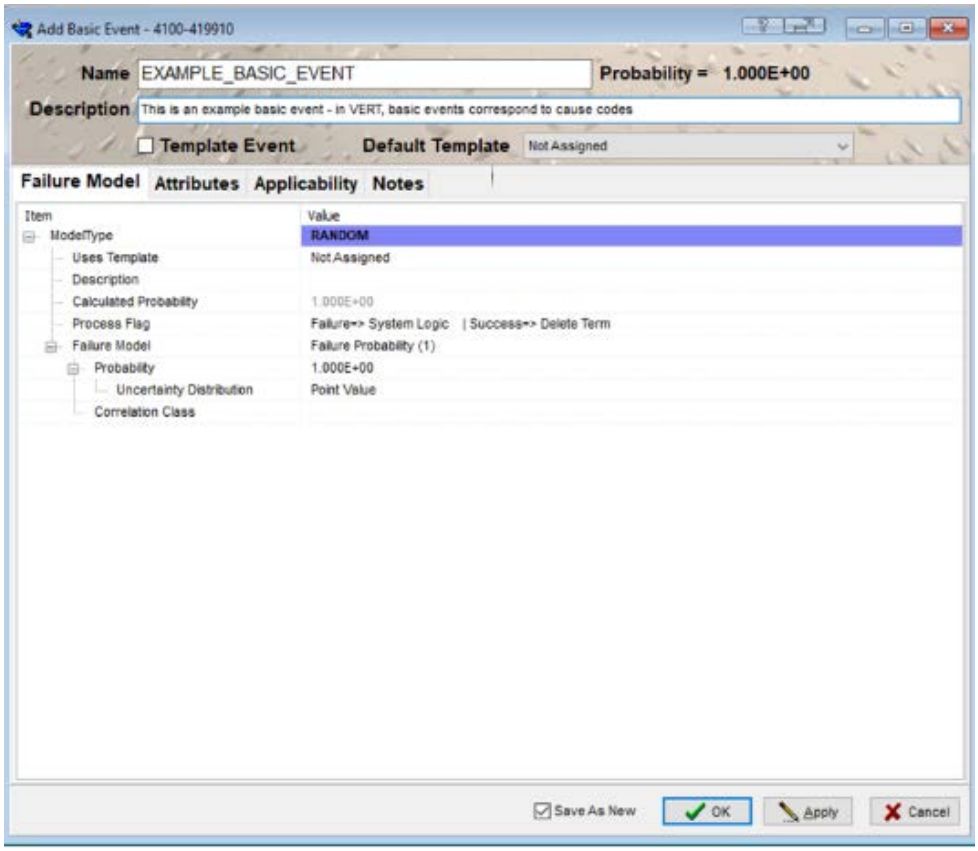


Figure 67. Basic event editing window.

Basic Event Report Generation

Generating a basic event report is an important step for VERT. To do this, first navigate to the basic event window. Next, from the dropdown menu, select “all” so all the existing basic events are displayed. Select one of the basic events and then click CTRL+A to select all the basic events. All the basic events should now be highlighted. Now select the “publish” dropdown box from the top of the main menu and click “Basic Event Report”. See Figure 68. In the next window, select the “Basic Event Listing” and “Excel Format (XLS)” options (see Figure 43), then select “Publish”. This will automatically open the basic event listing using a compatible program.

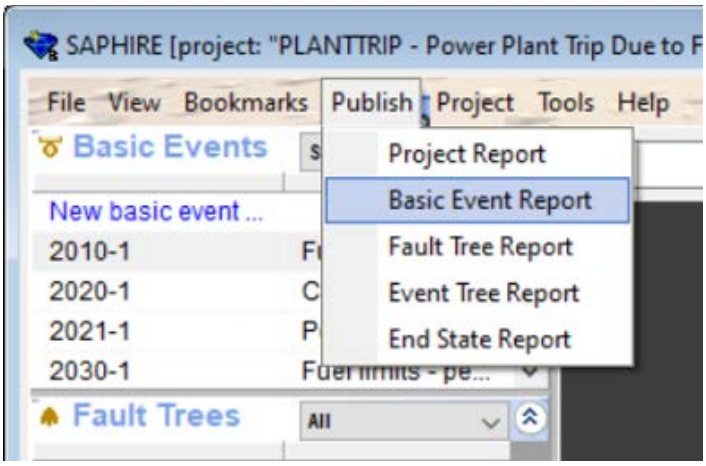


Figure 68. Basic event report location.

FT Rule Editing

Another important step for VERT is FT rule editing. To edit FT rules, click “project”, then “edit rules”, and finally “FT (Post-Processing)”. See Figure 69.

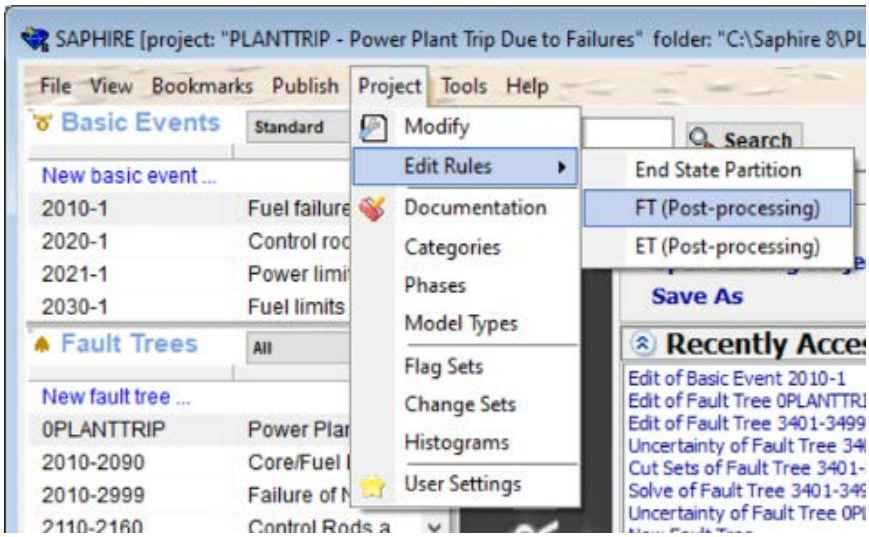


Figure 69. FT post-processing rules location.

A window similar to Figure 70 should appear. Enter the post-processing rules into the box, then click the “save” and “compile” buttons located at the top of the window under the “File” dropdown box.

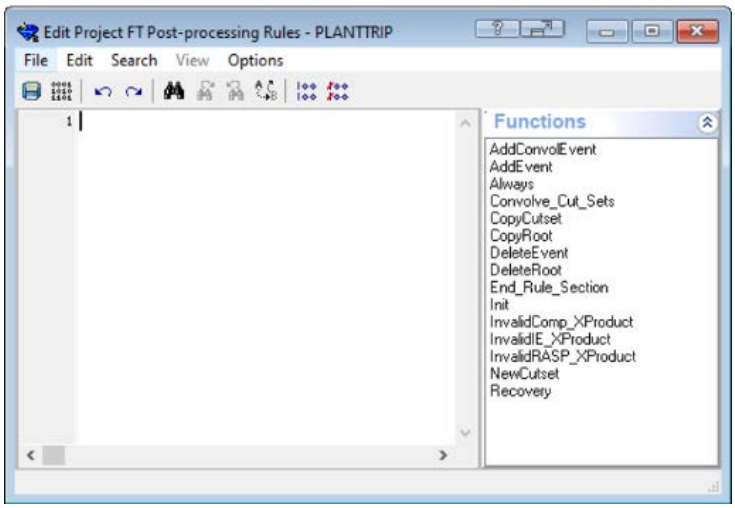


Figure 70. Post-processing rule editor window.

Solving FT

The final step is to solve the FTs. First select the "all" option from the dropdown box in the FT window in the main menu. Next, highlight all the FTs by selecting one and then pressing CTRL+A. Right click on a FT and select "solve". A window similar to Figure 71 should appear. In the "Solution Steps" section of the window, make sure to select the "Apply Post-Processing Rules" checkbox. Then click "Solve". Depending on the size and quantity of FTs, the solving process can take several minutes.

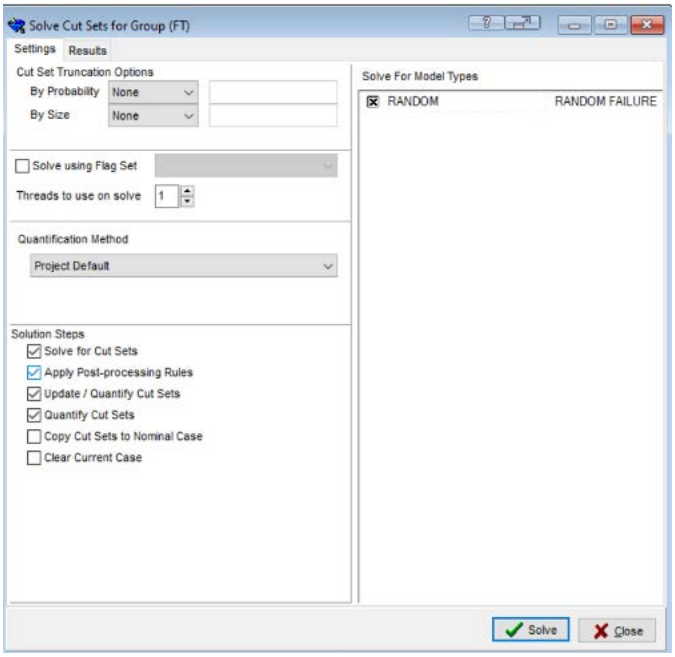


Figure 71. FT solve window.

APPENDIX E: INTEGRATING EXPERIMENAL DATA AND SIMULATION MODELS

E.1. Reactor Internals and Contact Wear

Of particular interest for this project is the degradation of reactor internals during the life of a reactor and from outage to outage. Due to the large flow velocities in a reactor, there can occur considerable vibration of the reactor internals. In this research, we are specifically interested in the core barrel and its support structures; see Figure 71. The core barrel motion can be of a variety of types: beam, tilting, shell, and others. The vibrations of the reactor internals are measured by fluctuations in the ex-core neutron detectors [14, 15, 16, 18, 20] (see Figure 72). Over time, the nature and contact between the reactor internals and the reactor vessel changes. A gradual decline in contact frequency as the unit ages is a result of a reduction in contact and an increase in the space between the reactor vessel and internals, e.g., wear in the radial key. Inspections at shutdown have supported the that interpretation that there is some "hammering in" of contact surfaces [20].

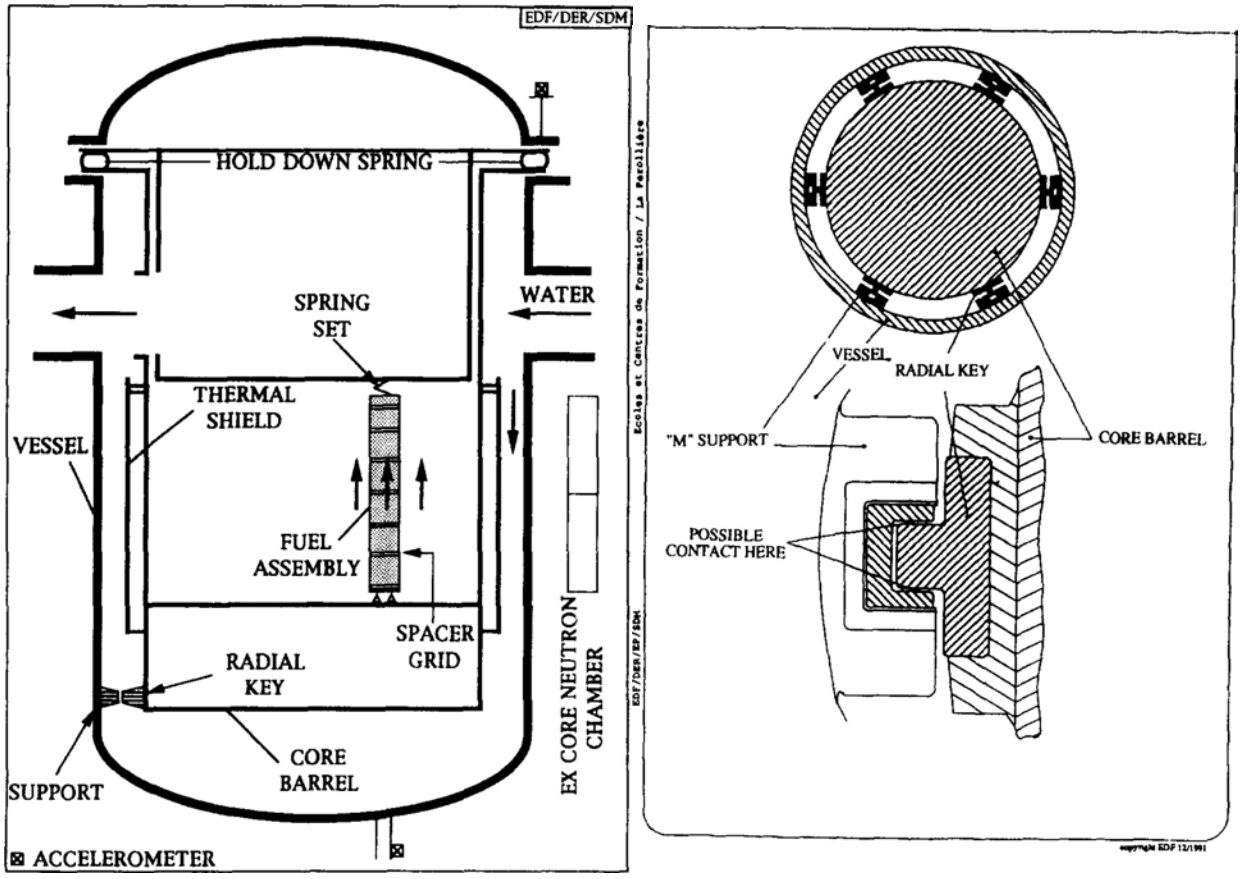


Figure 72. The core barrel is mounted in the reactor and held in place by hold-down springs. The motion at the base of the core barrel is constrained by the radial keys [20].

It is clear that changes in the contact and wear are indicative of the health of the core-barrel support structures. Some researchers have used the change in vibration signatures, measured by ex-core detectors, as an indicator of health, but this approach can incorrectly convolve changes in the vibrating structure itself

with changes, say, in the radial-key. Changes in the vibrating structure can result from factors other than contact at the radial key, while changes in the radial-key contact are a result of the "hammering in" phenomenon. Having an approach that lets us distinguish vibrations from the radial key is necessary.

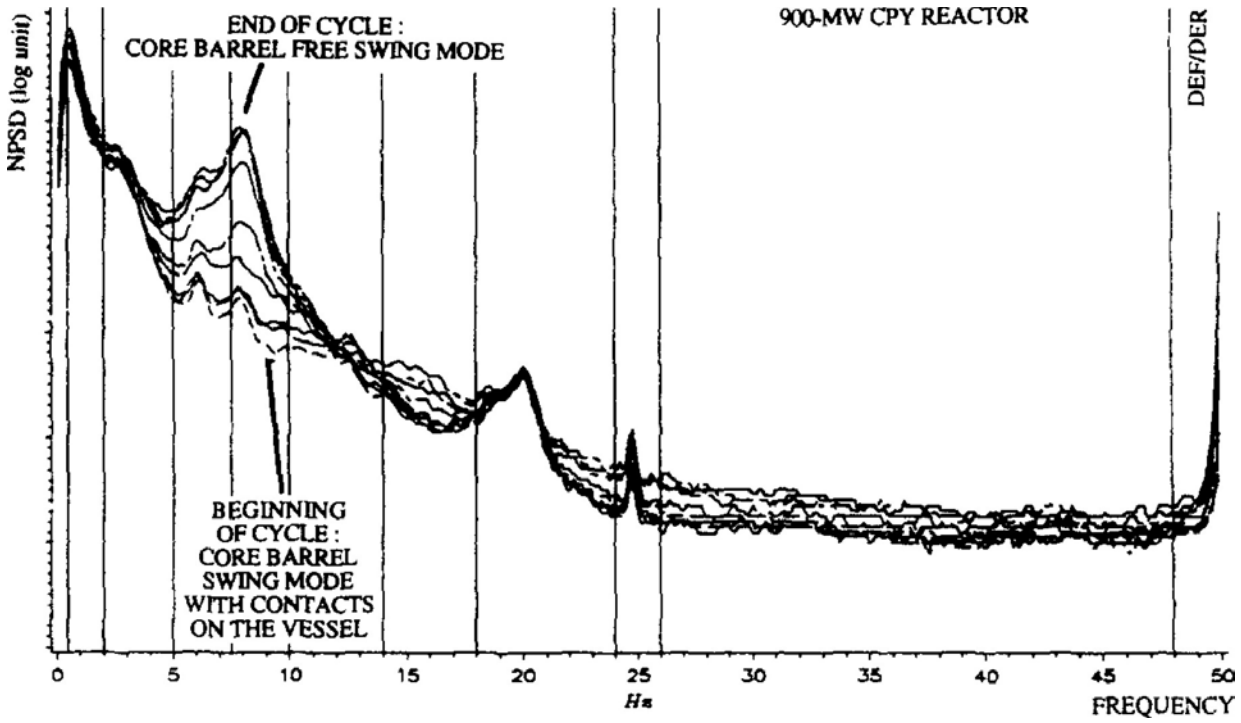


Figure 73. During the operation of the reactor, decreasing contact between the internals and the reactor vessel changes the signature pattern. Tracking changes over time is one method of monitoring the health of the core barrel, the radial key, and reactor internals. [source: 11]

What is challenging for this analysis is that the changing contact and wear often results from considerable changes in a non-linear contact. Trying to emulate the contact and characterizing the contact forces outside of the reactor in a laboratory environment is quite difficult. Trying to do so without the environment of the reactor, and without the resulting vibration of the reactor internals, would miss significant characteristics of the motion and forces needed to characterize the contact. Furthermore, determining the change in the contact is necessary to monitor the health of the support structures. During the life of the parts, continual vibrations create stresses on the parts that result in wear. This wear then changes the characteristics of the contact. Monitoring just the changes in the contact, versus changes in the vibrations of the reactor internals overall, provides a better picture of the health of the support structures like the radial-key. As such, an analysis of the contact forces must be done *in situ* and must provide a way to separate the contact forces from the vibration motion.

E.2. Integrating Model-Based and Data-Driven Techniques

Current data-driven approaches for component health use historical values for time to failure and establish inspection and maintenance based on that history. Other than periodic inspections, little about the operating condition of the component is used to establish its health. Recent developments use statistical characteristics of signals to establish residuals that can be tracked and changing trends can be used to signal

the need for inspection, repair, or replacement. These approaches are missing knowledge about the physical characteristics of the component and cannot provide any physical insight to the degradation mechanism. What is needed is an approach that can integrate physical knowledge based on well-modeled simulations and data-driven techniques that can account for the true operating conditions.

The challenge when integrating model-based and data-driven techniques is that there must be a coherent framework that combines the known physics-based model and the as yet unknown data-driven model of the degradation mechanism. What must be captured is the interaction between the physics-based model and the data-driven model. This interaction is a feedback mechanism where the response of a system drives degradation, and degradation then affects system response.

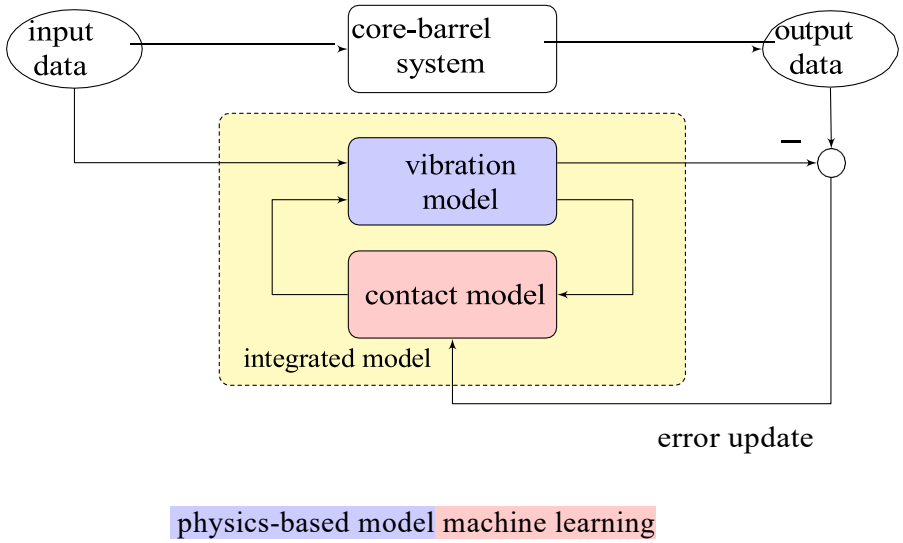


Figure 74. The interconnection of a vibrational system (model) and a contact mechanism is a feedback process. The contact mechanism can be identified using machine learning in real-time using model-matching. Its change over time can then be used to estimate and predict component health.

In this research, we explicitly model the feedback interaction between the nominal system and the degradation mechanism. Figure 73 illustrates this for our specific problem, which accounts for the effect of contact mechanics on a vibrational structure. This configuration is a common framework in systems and control theory, and the problem can be transformed into a standard model-matching problem where the objective is to determine, using data, the degradation mechanism that minimizes the error between the true system and the integrated model.

The advantage of the model-matching framework is that a variety of machine learning, parameter identification, and system identification techniques can be used. Importantly, modern data-driven techniques can be used to infer further physical laws for the degradation mechanism; among these are techniques for inferring solutions to differential equations using data [18], Gaussian processes [19], and deep learning of physical laws from data [11]. As discussed in Section E.4.2, we are using a full Bayesian approach to the model-matching problem. In this way, the uncertainty on all parameters of the integrated model (vibration plus contact models) and the uncertainty on the error will be characterized. Resolving the error uncertainty provides context for interpreting the integrated model's performance, which should aid stakeholders and decision makers.

Since the integrated model is a best match for the true system and explicitly includes the contact, by tracking the contact mechanism over time, we can track changes in the health of the true system. More importantly, once trained we can separate the contact mechanism from the system model, and we can elucidate the dependence of contact on operating condition. This provides new insight to the character of the contact in a way that *a priori* contact models cannot. This data-driven model can then be used for further hypothesis testing of the physics of contact and for forecasting health of components in real-time using real data.

E.3. Use and Effectiveness of the Machine Learning

An objective of this research is to use measurements of the core barrel motion, obtained from ex-core measurements, to infer the behavior and state of components within the reactor vessel. The measurements do not explicitly capture the response or state of the components. Changes in those signals reflect changes in the motion of the internal structures within the reactor vessel. The dynamic response of the internal structures may change throughout the life of the system as components degrade. Understanding the degradation and wear of internal structures is possible by collecting, processing, and analyzing the neutron noise signals throughout the cycle and across multiple cycles, as evidenced by previous studies.

Machine learning can be used to identify patterns and predict system behavior based on historical measurements. It is therefore natural to expect that machine learning techniques can be applied to this problem. However, machine learning strategies cannot be applied in a “black box” manner. To understand why, we need to consider how the data are collected and how the data are associated with the goal of the project. First, the measurements are indirect. The signals by themselves do not provide a direct target, class, label, or numeric value that represents the state of the structures within the reactor vessel. The signals will not allow an algorithm to find a relationship between the degradation of a component, such as the radial key inside the reactor vessel, because the data do not include any measurements inside the reactor vessel. Essentially, the input is the neutron noise signal and the output is the state of a component. The neutron noise data consist of zero observations inside the reactor vessel and so no direct input to output relationships can be established.

It might be tempting to consider using unsupervised learning approaches to overcome this limitation. Indeed, unsupervised learning makes no distinction between input and output. The goal is to find interesting patterns in the data, such as grouping or clustering observations together. Time series clustering, such as Dynamic Time Warping (DTW) or Principal Component Analysis (PCA) based approaches are common in machine learning research literature and commercial applications. Although those methods would identify groups of similar observations, those clusters would still suffer from lack of context. There would be no context about what physically makes the signals similar to each other. The goal is to understand the behavior inside the reactor vessel, but there would be no way of knowing which clusters correspond to degraded states versus nominal states. Worse yet, there would be no way of knowing which clusters contain a mixture of degraded and nominal conditions.

Ultimately the solution to these challenges is to bring in engineering context into the machine learning problem. Context can be provided several ways. In one approach, Subject Matter Experts (SMEs) could go through and manually provide context to the various signals. Such an approach could be viewed as a post-processing step to cluster analysis (an unsupervised learning technique). SMEs could then diagnose representative signals inside the identified clusters to provide the contextual clues about what makes the observations inside a cluster similar to each other and different from those in other clusters. This approach would combine traditional signal processing techniques executed by the SMEs with the automated cluster analysis of an unsupervised machine learning method. The downside of this approach is that it will be tedious and time consuming to examine the collected signals. As the SMEs study more signals, and thus

provide greater and more meaningful context, this approach would end up being as costly as first having the SMEs study and label the various signals before applying any machine learning methods. Either way, the approach will not scale well to studying behavior across years of data.

In a second approach, physical context is provided by numerical models. A numerical model makes use of the underlying governing equations, physical laws, and SME engineering insight to study the dynamics of the complex system. If the state of the system is known, the dynamics can be simulated, and the simulated response can be compared to the collected measurements. The simulation must capture the dynamical motion of the reactor internals and the neutron transport through the system to the ex-core detector. The initial system state and other related information may not be known, and thus must be learned. This inverse structure — given a response then learn the input — is aligned with the goal of the project where we wish to infer behavior indirectly through measurements of something else.

Bringing together computer simulations with the physical measurements is an area that has been researched, studied, and applied for decades. Point estimate "calibration" studies are used in numerous fields. Point estimates lack any aspect of uncertainty, since the results are single values associated with each unknown. Point estimates are not appropriate for this project because we wish to understand the state of a complicated system. It is imperative to have some representation of the uncertainty of the state, because a decision must ultimately be made to take some action. That action could be more thorough inspections, or even scheduling a replacement of the internal structures. That decision must consider the financial, logistical, and other management related impacts of performing such actions.

We are using a full Bayesian approach for inferring the unknown state by combining the neutron noise measurements with computer simulations. The computer simulations provide the context, and the full Bayesian approach represents the unknown states as probability distributions. Decisions can thus be made by considering the uncertainty in the state, which will help in considering the type of action that should be scheduled. For example, if the expected behavior is considered acceptable but the uncertainty is "large" an inspection could be scheduled. However, if the behavior is considered to be "precisely" near off-nominal conditions, then a replacement can be ordered and scheduled before a planned outage.

The full Bayesian approach will also aid the data collection and storage necessary to perform the inference task. The neutron noise data are high-frequency time-series signals. The data are streaming and continuously monitored during operation of the plant. However, long-term storage of such high-frequency signals may be challenging. Further, distributing such large volumes of data to the appropriate teams that will perform the inference task may be even more challenging. Data access, querying, and cleaning are the primary reasons that machine learning/data science projects fail in industrial applications (current estimates are between 60 % to 80 % of projects fail). Therefore, we cannot take the mindset of "give us all the data" and expect such a project to be successful in the long run. A limited data volume will be available, and the method must be capable of efficiently learning from the data. A full Bayesian viewpoint with appropriate prior distributions will be beneficial in learning from "small data" that ultimately must occur to solve this "big data" problem.

As stated in the previous section, we are proposing integrating machine learning techniques with physics models. Physical phenomena that are difficult to characterize and model will be approximated by machine learning methods, rather than governing equations or physical laws. Machine learning will therefore provide the necessary closure or constitutive relationships to model the behavior of the complex system. Machine learning and/or statistical models have been used to capture behavior outside the assumptions of physical simulations for two decades. The seminal work by Kennedy and O'Hagan (KOH) [13] used Gaussian processes (GPs) to capture the error or model discrepancy between simulations and measurements. The discrepancy function was learned in a full Bayesian context to capture as many sources of uncertainty as possible. Once learned, the discrepancy function enables the physics simulation to draw

conclusions from limited physical measurements, while the measurements enable the simulation to overcome the inherent simplifying modeling assumptions.

The KOH approach is a powerful technique, but it is challenging to implement. Computer simulations typically involve their own set of unknown parameters, which engineers hope to learn or calibrate using measurements. Learning the discrepancy function simultaneously with the unknown physical parameters is a challenging inference problem due to non-uniqueness, unidentifiability, and multi-modality [10,12]. Multiple authors have developed approaches to try and compartmentalize the various parameters as a way to simplify the learning problem. Ultimately, these challenges as a whole have made it difficult to implement the KOH approach in practice.

In this work, rather than trying to learn a general discrepancy function between the measurements and the physical simulation, we are using machine learning to represent specific physical processes. Machine learning is therefore still capturing difficult to model dynamics, but prior information about the behavior of such processes can be directly applied to regularize and constrain the machine learning models. As discussed earlier, the machine-learning approximated physical behavior can then be separated from the physics-based simulations. The machine learning model can be interrogated by SMEs to explore trends and behavior of the physical process.

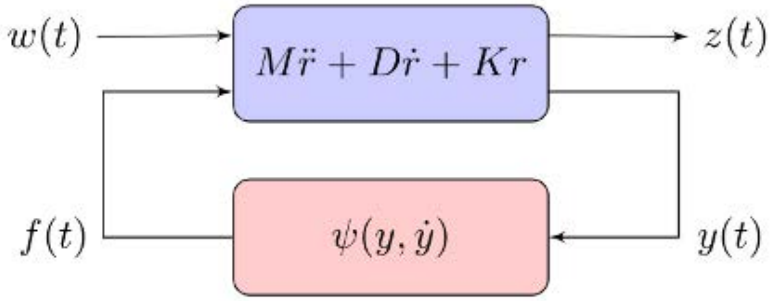


Figure 75. The interconnection of the vibration model of the core barrel and contact force is a generalized feedback problem.

E.4. Methods

E.4.1. Core-Barrel Model

E.4.1.1. Vibration Model

In this research, we isolate the vibrations from the contact by modeling the interaction between the vibrational system of the core barrel and the contact. The dynamics of the vibrating core barrel are assumed known and modeled in a vibrational model derived from finite element analysis, modal analysis, or similar approach. The unknown contact, that which we want to identify, will be characterized in a machine learning model. It is important to realize that the physics-based vibrational model and the machine-learning contact model are interconnected in some feedback process like that shown in Figure 73 and Figure 74. Mathematically we can describe the general vibrational model as:

$$M\,\ddot{r}(t) + D\dot{r}(t) + Kr(t) = B_1\,w(t) + B_2\,f(t) \tag{E.1}$$

$$z(t) = C_1\,r(t) \tag{E.2}$$

$$y(t) = C_2\,r(t) \tag{E.3}$$

$$f(t) = \psi(y(t), \dot{y}(t)) \tag{E.4}$$

The signals are:

- $\ddot{r}(t), \dot{r}(t), r(t)$: are the generalized displacement, velocity, and acceleration;
- $w(t)$ is an exogenous forcing function from the turbulent flow;
- $z(t)$ is the ex-core neutron measurement;
- $y(t)$ is the displacement at the point of contact; and,
- $f(t)$ is the force of contact.

And the matrices are:

- M is the mass matrix;
- D is the damping matrix;
- K is the stiffness matrix;
- B_1 is the input matrix for the exogenous forces;
- B_2 is the input matrix for contact forces;
- C_1 is the output matrix for the ex-core sensors;
- C_2 is the output matrix for the contact motion.

More specifically for this research, we are modeling a single mode of vibration with natural frequency of $\omega_n = 10\ Hz$. The exogenous force is sinusoidal with frequency of $\omega = 2\ Hz$ and stiffness normalized amplitude of $X = 2\ cm$.

E.4.1.2. Ex-Core Measurement Model

For the ex-core measurement, we assume the variation in the flux is

$$\delta\phi(t) \ = \ A\ exp(\Sigma\ r(t)) \tag{E.5}$$

where the constant A does not depend upon the perturbed position of the vibrational system and Σ is an effective macroscopic cross-section that describes the absorption by the core barrel of neutrons emitted from a radiation source (the core). Equation 5 represents a simplified age diffusion approximation to the thermal flux measured at the ex-core detector. The oscillatory behavior of the reactor internals therefore impacts the measurement by changing the effective distance between the source and the detector.

The ex-core measurement is the scaled variational log-normalized flux

$$\varphi(t) = a\ log\ \delta\phi(t) = \ a\ \Sigma\ r(t) \tag{E.6}$$

where the constant a describes the sensor calibration factor. In this problem, the constants a and Σ are arbitrary. We assume a measurement noise that is Gaussian, $n(t) \sim N(0, \sigma^2)$. The ex-core measurement is then:

$$z(t) = \varphi(t) + n(t) \sim N(a\Sigma r(t), \sigma^2) \tag{E.7}$$

E.4.1.3. Contact Force Model

We assume a non-linear friction force that is viscous (proportional to velocity) for small displacements from equilibrium and is constant for large displacements. We assume a hyperbolic tangent such that

$$f(t) = m\,\alpha\,tanh(\beta\,\dot{r}(t)) \tag{E.8}$$

The parameter α controls the asymptotic friction at large velocities; the parameter β defines the characteristic velocity $V_c = 1/\beta$. This model is a hybrid of linear viscous damping $f = c\,\dot{r}$ and constant Coulomb friction $f = F_0\,sgn(\dot{r})$ with $F_0 = m\alpha$. A graphical depiction of this is shown in Figure 75. With this assumption the equation of motion becomes

$$\ddot{r}(t) + \alpha\,tanh(\beta\dot{r}(t)) + \omega_n^2 r(t) = X\omega_n^2\cos(\omega t) \tag{E.9}$$

$$z(t) = a\,\Sigma\,r(t) + n(t) \tag{E.10}$$

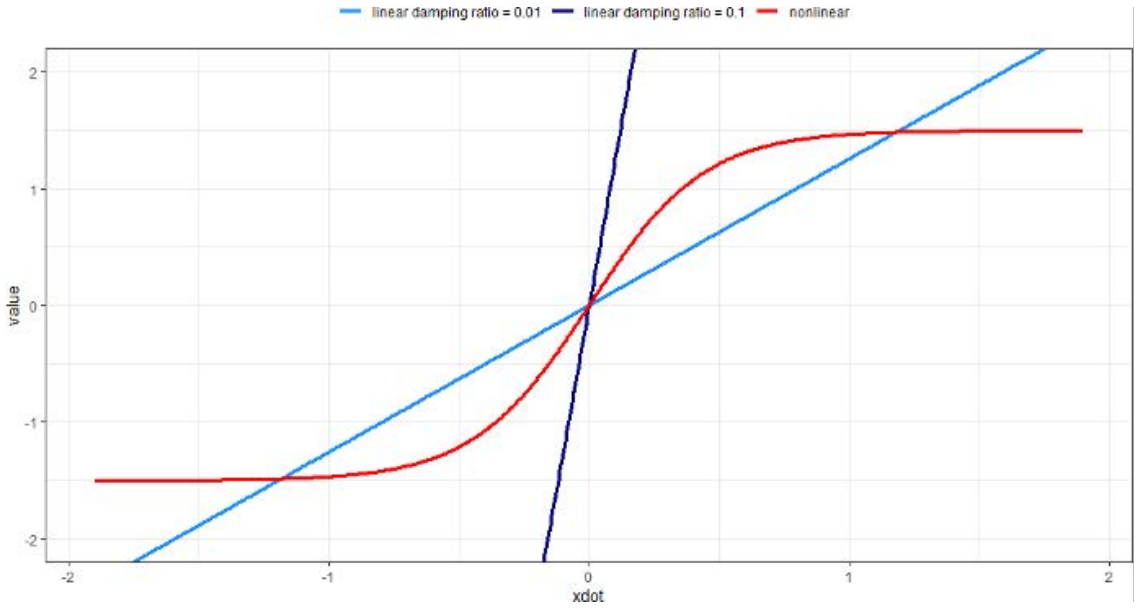


Figure 76. The friction force, shown in (red), is viscous (proportional to velocity) for small displacements from equilibrium and is constant for large displacements. The lightly damped cases of a damping ratio of 0.1 (light blue) and 0.01 (dark blue) are shown for comparison.

The α and β parameters control the behavior of the contact force, which impacts the evolution of displacement under an external load. The displacement impacts the ex-core measurements. The problem is to estimate the parameters α and β given the noisy measurement $z(t)$.

E.4.2. Inference Problem

Inference of the unknown machine-learning parameters, α and β, in a full Bayesian setting requires specifying the likelihood between the physics model and the measurements, as well as the prior on the unknowns. The prior may help prevent unphysical behavior by ruling out extreme values. The updated belief or posterior distribution on the unknown machine-learning parameters is a compromise between the data and the prior.

E.4.2.1. Likelihood Function

The data points correspond to N ex-core measurements collected at a specific data acquisition frequency. The measurement times are denoted as t_n where $n = 1, \ldots, N$. The vector of all N measurements is denoted as $z = \{z(t_0), \ldots, z(t_N)\}$.

The integrated model consists of two coupled physics simulation. First, the vibration dynamics are simulated by evolving the initial state forward in time. The machine-learning-based contact model is integrated with the vibration dynamics model. Second, the neutron transport phenomena must be simulated to represent how the ex-core measurement will evolve through time. The neutron transport requires the system displacement (the output of the vibration dynamics model) as an input. The combined result of these two models is the simulated flux at the ex-core detector. That response denoted as $\varphi(t_n)$, with all N responses denoted as $\boldsymbol{\varphi}$. The N measurements are assumed to be conditionally independent. Thus, the likelihood is the product the N conditionally independent likelihoods:

$$p(z|\boldsymbol{\varphi}, \sigma) = \prod_{n=1}^{N} p(z(t_n)|\, \varphi(t_n), \sigma) \tag{E.11}$$

Following the ex-core measurement model discussion earlier, each of the measurements is assumed normally distributed around the simulated response. The likelihood is thus the product of N conditionally independent Gaussian likelihoods:

$$p(z|\boldsymbol{\varphi}, \sigma) = \prod_{n=1}^{N} N(z(t_n)|\, \varphi(t_n), \sigma^2) \tag{E.12}$$

The likelihood appears rather simple because all of the time dynamics are captured by the physics simulation. The learning problem would be considerably more complex without the physics simulation because the autoregressive nature of the response would need to be captured within the likelihood directly.

E.4.2.2. Prior Specification

As discussed in Section E.4.1.3, the contact force is a hyperbolic tangent function, an "S-curve".

$$f(t) = m\, \alpha\, tanh(\beta\, \dot{r}(t)) \tag{E13}$$

In this section, we are focusing on the specification of the machine-learning parameters, α and β. The full prior consists of the joint prior distribution on both parameters, $p(\alpha, \beta)$. We take the approach that a priori both parameters are independent,

$$p(\alpha, \beta) = p(\alpha)\, p(\beta) \tag{E.14}$$

Appropriately specified informative prior distributions on the α and β parameters may help prevent nonphysical behavior of the contact force. For example, the hyperbolic tangent model converges to a constant Coulomb friction model as the β parameter increases in absolute value. If it is believed, a priori, that the contact force should depend on the velocity, then the prior can be formulated to prevent large β values. Uninformative or vague priors would be susceptible to noisy observations, and so may result in incorrect behavior in small sample sizes. Thus, this research uses moderately regularizing priors on α and β to try and prevent large values of either parameter. For simplicity, Gaussian priors are used for both parameters. The priors are parameterized with prior means, μ_a and μ_b, and prior standard deviations, τ_a

and τ_b, on the α and β parameters, respectively. Using this notation, the joint prior can be written as the product of two independent Gaussians:

$$p(\alpha, \beta) = N(\alpha|\mu_a, \tau_a^2)\, N\left(\beta\middle|\mu_b, \tau_b^2\right) \tag{E.15}$$

The prior means are both set to 0, $\mu_a = \mu_b = 0$, and the prior standard deviations are both set to 3, $\tau_a = \tau_b = 3$, to moderately regularize the parameters. This prior formulation is analogous to using a ridge penalty in a non-Bayesian setting.

E.4.2.3. Prior Specification

The posterior distribution is proportional to the product of the likelihood and the prior. In this case, we are updating our belief about the α and β parameters given the N measurements, $z(t_n)$. The simulated response, $\varphi(t_n)$, is a function of the machine-learning parameters. Thus, $\varphi(t_n)$ can be viewed as a non-linear mapping between the machine-learning parameters and the expected value for the measurement at a given point in time:

$$E\{z(t_n)|\alpha, \beta\} = \varphi(t_n, \alpha, \beta) \tag{E.16}$$

The simulated response at time t_n will be written in a more compactly as, $\varphi_n(\alpha, \beta)$ for the remainder of the appendix. Using this notation, the n-th likelihood can be written as:

$$p(z(t_n)|\alpha, \beta, \sigma) = N(z(t_n)|\varphi_n(\alpha, \beta), \sigma^2) \tag{E.17}$$

Using this notation, the posterior distribution on α and β given the N measurements is written as:

$$p(\alpha, \beta|z, \sigma) \propto \prod_{n=1}^{N} N(z(t_n)|\, \varphi_n(\alpha, \beta), \sigma^2)\, N(\alpha|0, \tau_a^2)\, N\left(\beta\middle|0, \tau_b^2\right) \tag{E.18}$$

The posterior distribution in this equation holds as long as there are just two unknown machine- learning parameters and that the measurement error, σ, is known. The physics simulation, $\varphi_n(\alpha, \beta)$, could be simple analytic result or a complex, long running simulation.

E.4.2.4. Estimating Posterior Distribution

The hyperbolic tangent contact model discussed throughout this appendix consists of two unknowns, α and β. The posterior distribution of such a small dimensional problem can be solved directly via graphical methods. The graphical solution is sometimes referred to as the direct method or grid approximation, because the posterior can be evaluated at a grid of many possible "candidate" parameter values. A benefit of such an approach is that the only approximations in the solution scheme is the decision for how many grid points to use, which thus control the resolution of the posterior "image". The main steps of the grid approximation are

1. Specify a grid of candidate parameter values, $(\alpha, \beta)_m$ where $m = 1, \ldots, M$
2. For each of the $m = 1, \ldots, M$ grid points:
    a. Integrate the physics simulation forward in time using the specific set of machine-learning parameters, $\varphi_n((\alpha, \beta)_m)$ where $n = 1, \ldots, N$
    b. Calculate the log-posterior, $log\ p((\alpha, \beta)_m|\, z, \sigma)$
3. Visualize the posterior surface over the candidate parameter grid

The candidate grid is usually set based on the prior. The grid approximation does not scale well beyond 2 parameters; however, it does provide a graphical tool to understand the behavior of the "true" solution, the posterior distribution without any approximations. It thus helps diagnose challenging behavior, albeit for small-scale applications.

To scale beyond two unknowns, rather than using MCMC sampling, this research chose to use a distributional approximation to the posterior. Specifically, the Laplace approximation (quadratic or normal approximation) was applied. The Laplace approximation is derived from a second-order Taylor series expansion of the log-posterior function (see Equation E.18) around the posterior mode. The Laplace approximation approximates a joint distribution as a multivariate normal (MVN) distribution. The mean of the MVN is set equal to the mode of the joint distribution and the covariance matrix is calculated using the local curvature around the mode. The Laplace approximation is thus a local approximation to the uncertainty in the parameters.

The benefit of the Laplace approximation is that it is relatively fast to execute. An optimization algorithm is used to find the posterior mode. Compared to MCMC sampling, a relatively small number of iterative evaluations are required. The Hessian matrix was evaluated numerically, and thus only needed to be estimated after the posterior mode was identified.

E.5. Results and Discussion

In this section, we examine the ability to recover the parameters that were used to generate the data in order to understand potential pitfalls and solution strategies to overcome those challenges. The physics simulation was integrated forward in time from an assumed initial condition, and noisy measurements were generated. The true values for the machine-learning parameters were specified to produce the contact force model displayed as the red curve in Figure 75. The goal of the parameter recovery demonstration was therefore to see if the inference engine could learn the parameter values that generated the data. The grid approximation and the Laplace approximation strategies were applied.

E.5.1. Inference of a Non-Linear Loss Model

E.5.1.1. Grid Approximation Inference

The grid approximation was applied first to graphically solve the joint posterior on the unknown contact force parameters, (α, β). Both parameters have assumed Gaussian prior distributions. A grid of candidate points was created as a 115 × 115 full-factorial grid between ±3 prior standard deviations around the prior mean. The log-posterior was calculated for all 13225 combinations over the grid.

The grid approximation results are displayed in Figure 76 below. The α parameter is displayed on the horizontal axis and the β parameter is displayed on the vertical axis. The surface fill color represents the posterior density associated with each grid point (α, β). Yellow areas have the highest posterior density, dark blue regions have very low posterior density, and all (α, β) considered implausible are greyed out. In Figure 76, implausible is defined as having a posterior probability of less than 0.01 %. Posterior contours are displayed by the black curves. The graph has been zoomed in between -6 and +6 for both parameters. The most striking aspect of Figure 76 is that most of the combinations of the two parameters have been "greyed out". The second major take away is that the distribution is bimodal having a significant "banana" shape; this bimodal posterior shape makes sense because the hyperbolic tangent is an odd function. One mode corresponds to positive parameters, and the second corresponds to negative parameters.

This particular contact force model is a simplified single hidden layer feedforward neural network with a single neuron and a hyperbolic tangent activation function. The bias parameters of the neural network have been set to 0 and thus this simple neural network consists of only unknown weight parameters. Neural networks are notorious for having complex multi-modal objective functions (specifically here we are investigating the posterior surface which includes the effect of a prior). Machine learning models integrated within physics simulations will also suffer from these challenges, as visualized by Figure 76.

Earlier in this appendix, it was stated that SME opinion can be easily incorporated into the framework by applying the machine-learning model directly to physical processes. For example, if SMEs believe that the contact force should increase as the velocity increases a natural constraint is to assume that the parameters should both be positive. Figure 77 zooms in on the quadrant with both positive parameters. The thin "banana" shape is apparent, representing a hyperbolic relationship between the two. It is difficult to assess if there is a single mode within this thin region because of the resolution of the grid approximation. Increasing the resolution of the grid approximation in this region would be expected to resolve such behavior.

However, rather than applying the grid approximation just in the positive-positive quadrant, the desired lower bound constraint on the two parameters can be applied. Both parameters are log-transformed and thus the posterior distribution can be reformulated to study joint distribution of the log-transformed parameters,

$$\tilde{\alpha} = log\ \alpha,$$

$$\tilde{\beta} = log\ \beta,$$

rather than the original parameters. This transformation has modified the priors to be log-normal priors on the log-transformed parameters. Applying the transformation allows the original Gaussian priors centered at zero to still be used which simplifies the prior specification process.

The grid approximation was executed for the log-transformed parameter space. The steps are the same as in the original parameter space. The candidate grid was setup as 125 ×125 full factorial combinations between log(0.5) and log(10.5) for both transformed parameters. The result is shown in figure 8. The hyperbolic like thin "banana" shape is still present, but it is less pronounced in the log-transformed space. Most of the possible combinations of the two log-transformed parameters are still ruled out, representing that only a narrow "ridge" of parameters is plausible. With the higher image resolution, it is easy to see that a single mode exists. SME subjective opinion has therefore made a challenging neural network fitting operation more tractable by removing the second mode.

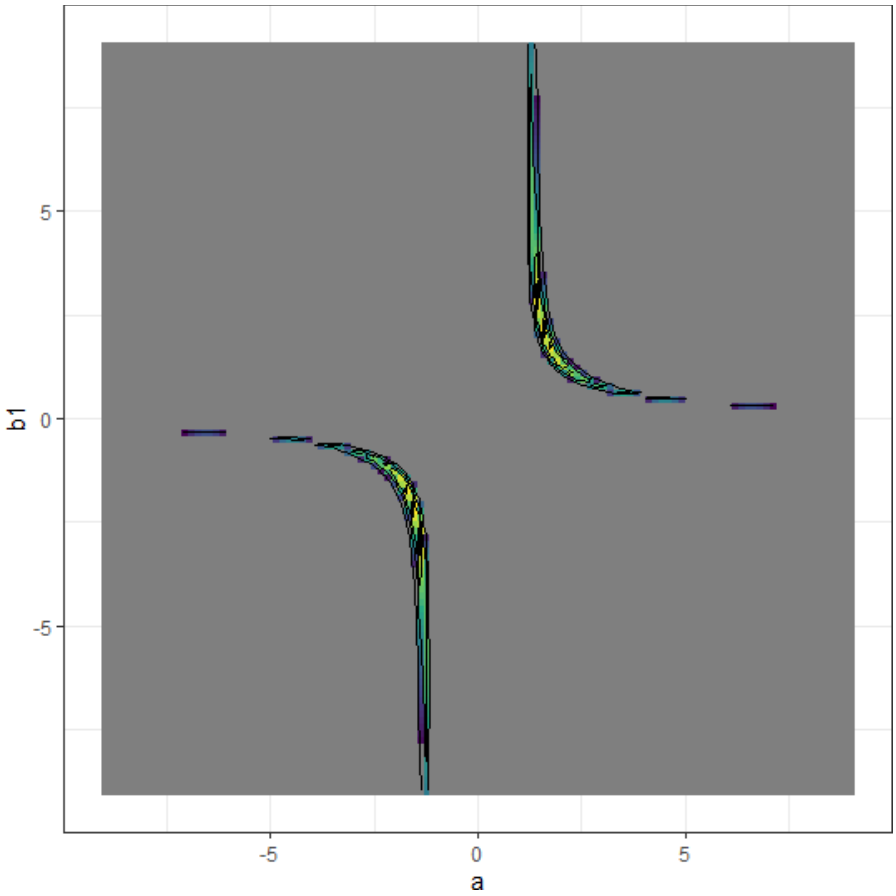


Figure 77. Joint posterior surface graphically solved by the grid approximation for the unknown machine-learning parameters α and β.

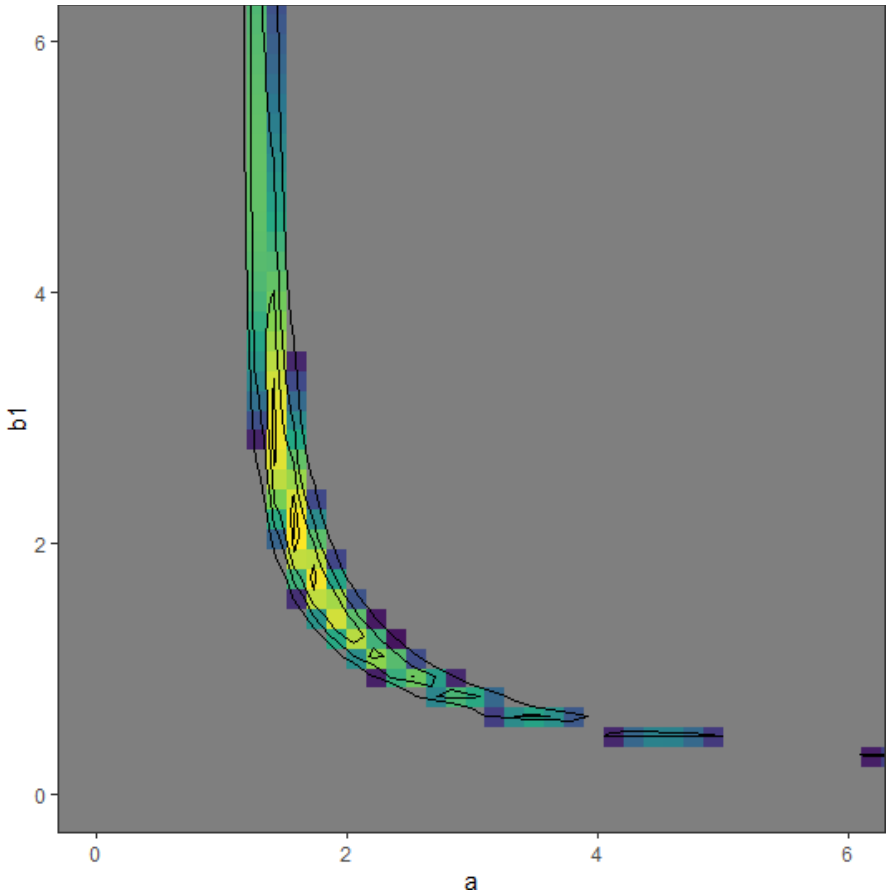


Figure 78. Zoomed in joint posterior surface grid approximation on the positive machine-learning parameter combinations.

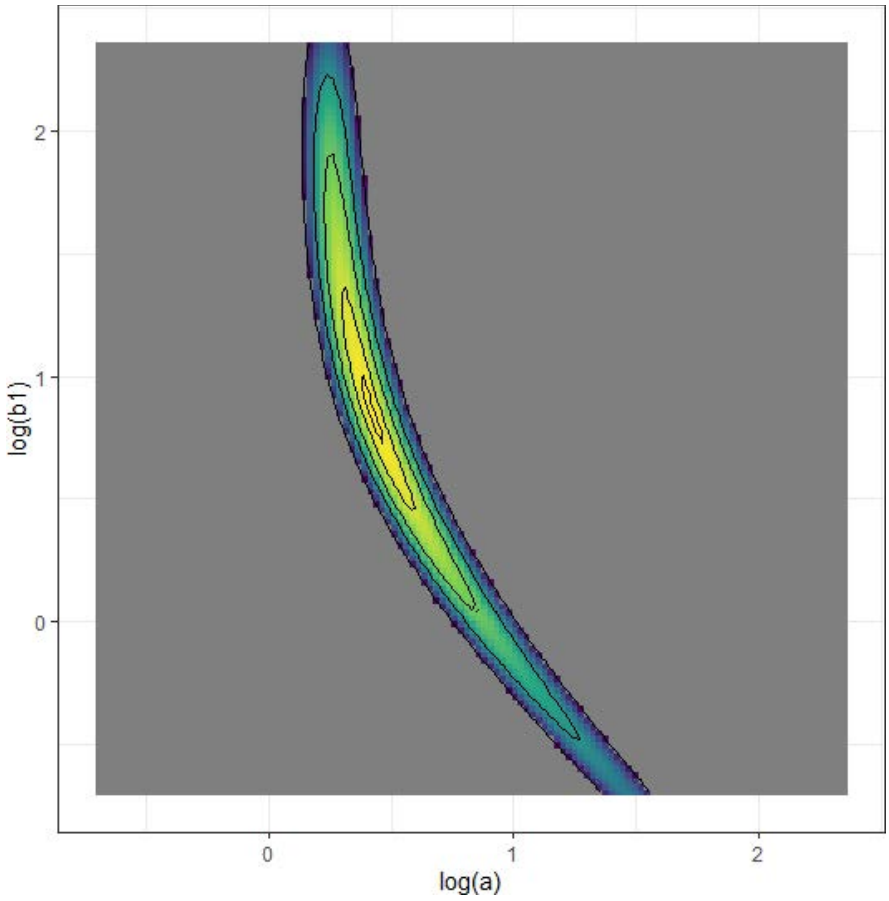


Figure 79. Joint posterior surface graphically solved by the grid approximation for the log-transformed machine-learning parameters, log (α) and log (β).

E.5.1.2. Laplace Approximation Inference

With the true posterior surface visualized, we apply the Laplace approximation to understand the limitations of the approximation. The MVN distribution is unimodal and so the Laplace approximation is appropriate for the log-transformed parameter space. The gradient and Hessian matrix calculations were executed numerically. Because the log-transformed space is unimodal we should not expect the initial guess values for an iterative gradient based optimizer to impact the final results.

The Laplace approximation to the log-transformed joint posterior is compared relative to the true log-transformed posterior in Figure 79. The posterior surfaces are visualized in the same manner as the previous figures, except that only the contours are displayed. The black contours denote the true posterior and the orange contours denote the Laplace approximation result. The posterior is well approximated around the posterior mode. However, the approximation struggles in the extreme tails since the MVN is not capable of representing the "banana" curves to the surface. This is a limitation to the approach, but we see that the method captures the behavior around the most probable values quite well. Overcoming this limitation requires either variational inference techniques, which assume a different distribution to the posterior (potentially as a mixture model), or with MCMC sampling.

Ultimately, we transform the log-transformed variables back to the original parameter space. Random sampling is used to accomplish this. An MVN random number generator is used with the mean equal to the mode and covariance matrix set to the results from the Laplace approximation. A desired number of

posterior samples are generated with all samples being independent of the others. The original parameter posterior samples are calculated by applying the inverse transformation functions.

The posterior marginal histograms of the machine-learning parameters are displayed in Figure 80. Also show is a posterior scatter plot of the parameters. As reference, the true parameter values used to generate the data are displayed as vertical red lines on the histograms and as the red dot in the scatter plot. The posterior modes for α and β line up well with the true parameter values. The posterior uncertainty in both parameters appears rather high, however, the parameters are quite anti-correlated. This is due to the fact that the hyperbolic tangent function can increase due to both α or β increasing. The two parameters can trade-off each other while still maintaining the same behavior of the contact force.

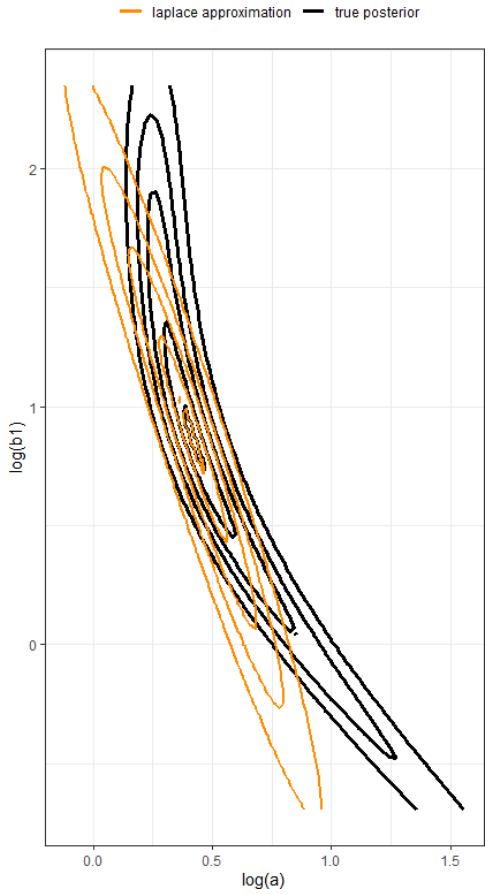


Figure 80. MVN approximate posterior relative to the true posterior for the log-transformed machine-learning parameters, log (α) and log (β).

E.5.1.3. Inference of the Contact Model Parameters

The machine-learning parameters control the behavior of the contact force. Thus, the posterior distribution on α and β can be propagated forward through the hyperbolic tangent function to investigate the posterior predictive distribution on the contact force. The inverse problem learns the parameter values, and then a forward problem allows studying the behavior of the machine learned physical process.

The posterior predictive distribution of the contact force with respect to the velocity is displayed by Figure 81. The black curve is the posterior predictive mean trend. The inner most dark grey ribbon is the

posterior middle 50 percent uncertainty interval (between the 25th and 75th percentiles) and the outer grey ribbon displays the posterior middle 90 % uncertainty interval (between the 5th and 95th percentiles). For reference, the true contact force assumed in this problem is displayed by the dashed red curve. The contact force behavior is well learned with the "S-curve" shape easily identifiable in the posterior predictive distribution. The uncertainty seems quite low relative to the uncertainty in the parameters due to the strong anti-correlation between the two parameters. The machine-learning parameter posterior represents a balance or trade-off between α and β, which correspond to many similar contact force models. The posterior predictive distribution displayed in Figure 81 represents that trade-off.

The machine-learning model can thus be isolated from the complete integrated model. In this demonstration problem that isolation represents studying the behavior of the contact force as the velocity changes. A more complex situation would involve studying the learned dynamics under various assumptions without having to run the entire physics simulation over and over again.

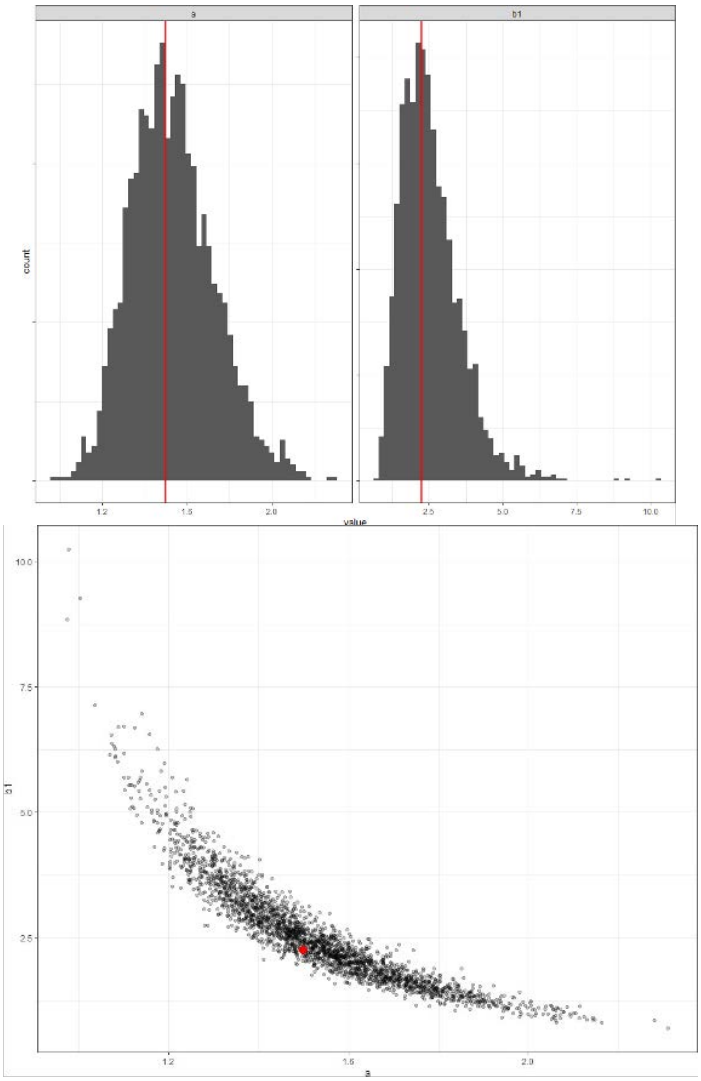


Figure 81. The posterior marginal histograms of the machine-learning parameters α and β; the red lines are the true parameters (top), and the posterior scatter plot of the parameters; the red dot is the true parameter value (bottom).

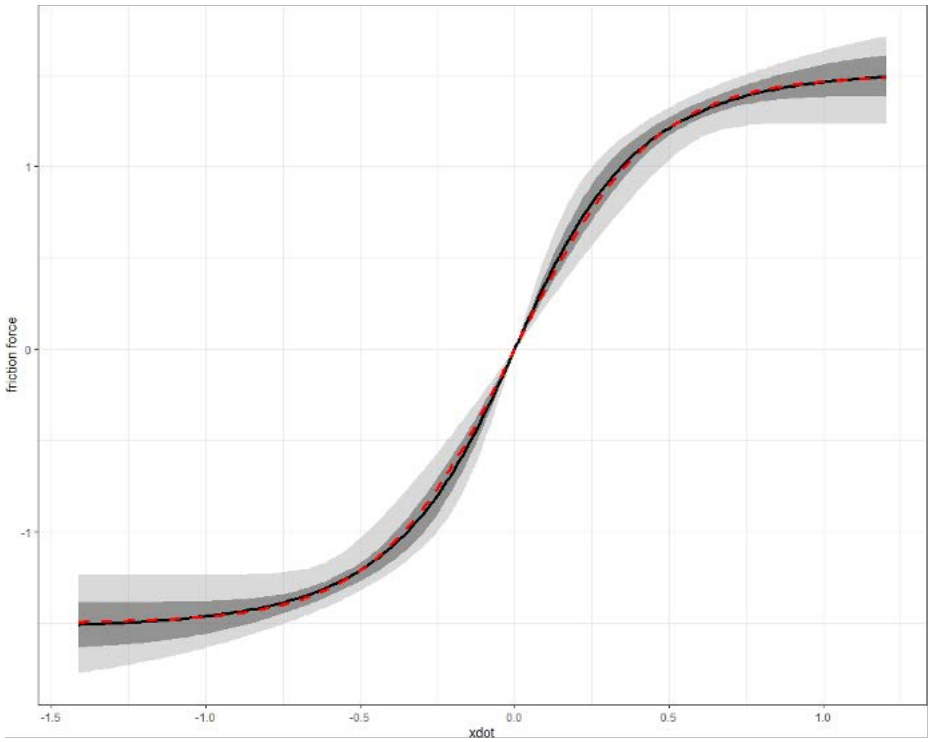


Figure 82. Posterior predictive distribution on the learned contact force with respect to the velocity.

E.5.2. Selection of the Contact Force Model

The previous sections performed the parameter recovery exercise using the exact model formulation that generated the data. Doing so is an important first step in studying the behavior of the posterior under different prior assumptions and sample sizes. A natural question to ask, what if a more general neural network formulation was used? Instead of using a single layer feedforward neural network with the bias parameters set to zero, what if the bias terms were estimated along with the weights? This section looks at how we can select amongst various data-driven model.

The contact force model with the general single hidden layer and single neuron neural network has a total of four unknown parameters

$$f(t) = m(\alpha_0 + \alpha_1 tanh(\beta_{0,1} + \beta_{1,1}\,\dot{r}(t))) \tag{E.19}$$

The biases, α_0 and $\beta_{0,1}$, are the intercepts of the output layer and the hidden linear unit, respectively. The weights, α_1 and $\beta_{1,1}$, are the slopes of the output layer and the hidden linear unit, respectively. All four machine-learning parameters must be learned simultaneously. The neural network will be susceptible to multi-modal posteriors, as described earlier. Unfortunately, it might be more difficult to impose constrains on the general formulation to try and remove the multiple modes. After all, simplifying the problem to just positive parameters and no bias terms reduces to the formulation described previously.

This more complex model still consists of a single hidden unit (neuron) and would be considered "simple" by machine learning standards. In a real application, it is tempting to use as complex a model as possible. More parameters must be learned, as more hidden units (neurons) and hidden layers are used. For example, a single layer two hidden unit neural network model for the contact force is:

$$f(t) = m(\alpha_0 + \alpha_1\, tanh(\beta_{0,1} + \beta_{1,1}\,\dot{r}(t)) + \alpha_2\, tanh(\beta_{0,2} + \beta_{1,2}\,\dot{r}(t))) \tag{E.20}$$

Neural networks, especially deep networks, are quite easy to overfit to training data, where the model becomes highly sensitive to small changes in the training set. Cross-validation is commonly used in machine learning to try and guard against overfitting, by repeatedly splitting the complete data into randomly selected training and hold-out test sets. A training/test split is referred to as a fold, and within each fold the model is trained on the training subset and then validated on the hold-out subset. This process is repeated for the desired number of folds. The generalization error is approximated by averaging the performance over the folds.

We cannot use such an approach in our current application. There is a natural order to the data, and so randomly selecting the training and test splits would be inappropriate. Time series cross-validation methods ensure the hold-out sets are always structured as true forecasts, where the trained model is always evaluated against future behavior. That said, appropriately validating a complex time series model is difficult. Information criterion based performance metrics, such as AIC and BIC in non-Bayesian settings, DIC and WAIC in full Bayesian settings, are common alternatives which penalize complex models based on their number of parameters. Information criterion simplify the model selection process in that all of the data are used to train the model.

We will use an information criterion like approach to compare the model performance. In addition to using all of the data, our choice of the Laplace approximation lends itself quite naturally to comparing models this way. The posterior distributions discussed in Section E.4.2.3 were formulated as proportional to, rather than as the complete Bayes' theorem. That is because for most computational Bayesian implementations, the denominator of Bayes' theorem is simply a normalizing constant, and can be ignored. Using the formulation for the general posterior of the set of $\boldsymbol{\xi}$ machine-learning parameters, the "complete" posterior distribution statement is:

$$p(\boldsymbol{\xi}|\boldsymbol{z},\sigma) = \frac{p(\boldsymbol{z}|\varphi(\boldsymbol{\xi}),\sigma)\,p(\boldsymbol{\xi})}{\int p(\boldsymbol{z}|\varphi(\boldsymbol{\xi}),\sigma)\,p(\boldsymbol{\xi})\,d\xi} = \frac{p(\boldsymbol{z}|\varphi(\boldsymbol{\xi}),\sigma)\,p(\boldsymbol{\xi})}{p(\boldsymbol{z},\sigma)} \tag{E.21}$$

where φ(ξ) denotes the vector of simulated responses as a function of all unknown machine-learning parameters. The denominator integrates the likelihood over the prior and thus averages the predictive density relative to the prior density. The denominator therefore accounts for all possible values the unknown parameters can take. Values that allow the model to match the data well, as well as values that lead to large errors. The denominator, which is known as the marginal likelihood or evidence, therefore accounts for the balance between predictive performance and complexity since complex models will have many possible combinations of the parameters which do not fit the data well. Comparing models based on the marginal likelihood therefore attempts to find the simplest model that explains the data. The marginal likelihood is difficult to calculate. However, the Laplace approximation provides a simple estimate. Thus, we can compare models based on their approximate marginal likelihood as a way to penalize overly complex models and therefore find the simplest model. This approach was discussed in terms of an information criterion metric because the BIC (Bayesian information criterion) is an approximation to the Laplace approximation's estimate to the marginal likelihood.

A common strategy for comparing Bayesian models with their approximate marginal likelihoods is to calculate the Bayes factor, which is the ratio of one model's marginal likelihood to another. In our application, we are interested in comparing the neural network contact force model with one hidden unit, ψ_{NN1}, to the neural network contact force model with two hidden units, ψ_{NN2}. The Bayes factor is calculated as:

$$BF = \frac{p(\boldsymbol{z}|\sigma,\psi_{NN1})}{p(\boldsymbol{z}|\sigma,\psi_{NN2})} \tag{E.22}$$

A Bayes factor value above one means that there is greater evidence for the ψ_{NN1} compared to the ψ_{NN2} model. A general rule of thumb is that if the Bayes factor is greater than 10, then there is strong evidence to support ψ_{NN1}, while values less than 5 are considered that the two models are relatively similar.

The same synthetic data used in the parameter recovery demonstration in Section E.5.1 are used to demonstrate model selection between more general neural network formulations. The single hidden unit contact force and the two hidden unit contact force models are considered because the data were generated by a simpler version of the single hidden unit model. Thus, the goal is to confirm that the most appropriate model is the simpler of the two.

The major steps for performing model selection are similar to those for applying the Laplace approximation. However, because we are dealing with general neural network formulations it is more difficult to consider how to properly constrain the parameters to ensure there is just one posterior mode. For those reasons, moderately regularizing zero-mean Gaussian priors with prior standard deviations of 3 are used for all parameters. Since the Laplace approximation cannot handle a multi-modal surface, the optimization is performed from multiple random initial guess values for each model type. The result with the highest marginal likelihood estimate is selected as the best for each model type and those two models are then compared relative to each other.

Before comparing the single hidden unit to the two hidden unit contact force model, the marginal posterior distributions are summarized via box-plots for each of parameters in the single hidden unit model in Figure 82, and likewise Figure 83 displays the summaries on the two hidden unit model. The single hidden unit model has 4 parameters, while the two hidden unit model has 7 parameters. In both figures, the box-plot colors denote whether the parameter is associated with the output layer or hidden unit of the neural network contact force model. Inspection of Figure 82 reveals that the bias terms (denoted as alpha_0 and beta_0_1 in the figure) are close to zero. The posterior correlations are not displayed, but many of the two hidden unit neural network parameters have relatively high posterior correlation.

The Bayes factor between the single hidden unit and two hidden model was calculated using the Laplace approximation estimates to the marginal likelihood.

$$BF = \frac{p(\boldsymbol{z}|\sigma, \psi_{NN1})}{p(\boldsymbol{z}|\sigma, \psi_{NN2})} \cong 8.75 \tag{E.23}$$

Following the guidelines, there is substantial evidence to support the simpler single hidden unit neural network, relative to the more complex two hidden unit model. The Bayes factor is not greater than 10, so this implies that there is still some posterior chance that the two hidden unit model could be correct. The posterior model weights are calculated and displayed in Figure 84, which show that there is roughly a 90% chance that the single hidden unit model is correct.

For further context as to why, the posterior predictive distributions on the contact forces with respect to the velocity are shown in Figure 85. The more complex two hidden unit model captures the correct trend; but is much more uncertain compared to the single hidden unit model. The high posterior predictive uncertainty reflects the fact there many different ways to combine the parameters of the two hidden unit model to capture the general behavior in the data. There might be a particular posterior mode that enables the more complex model to match the data very well, but we would not expect that model to perform as well as the simpler model on new data.

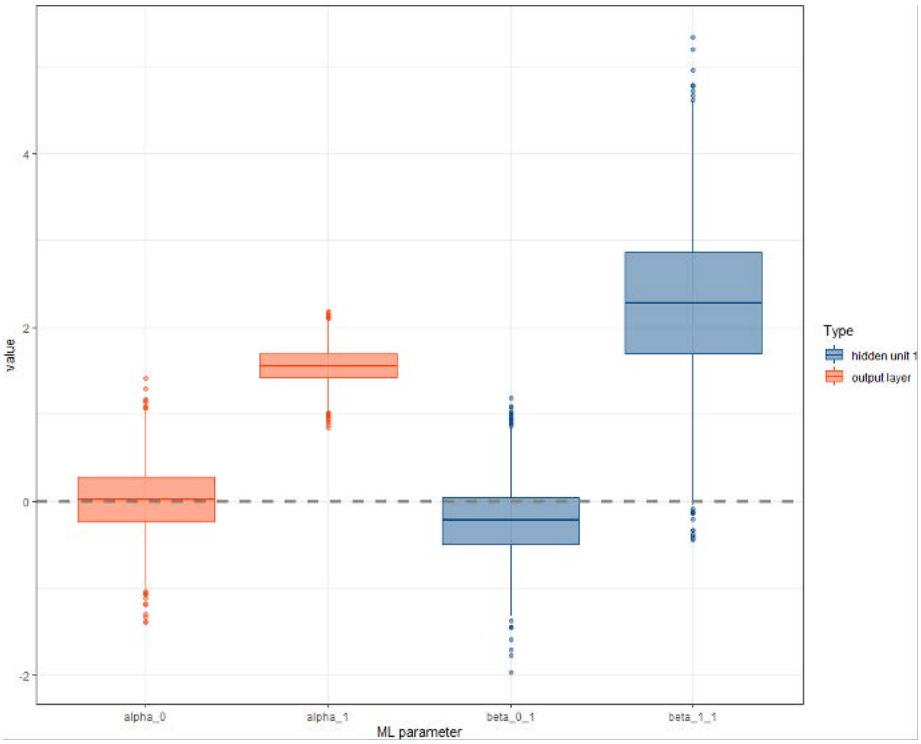


Figure 83. Marginal posterior parameter box-plots for the parameters of the single hidden unit neural network model.

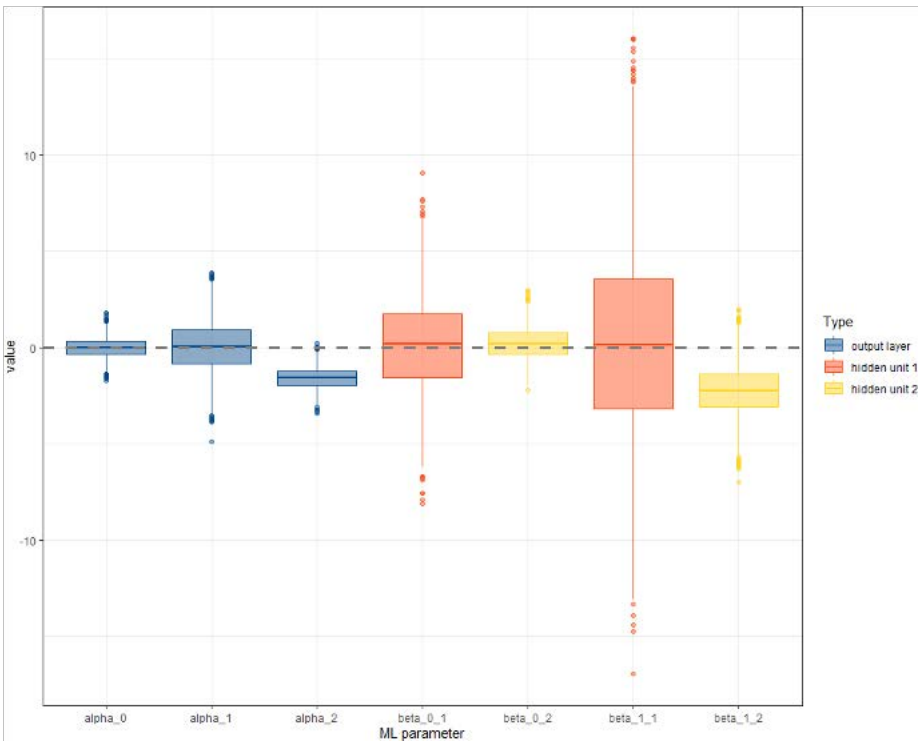


Figure 84. Marginal posterior parameter box-plots for the parameters of the two hidden unit neural network model.

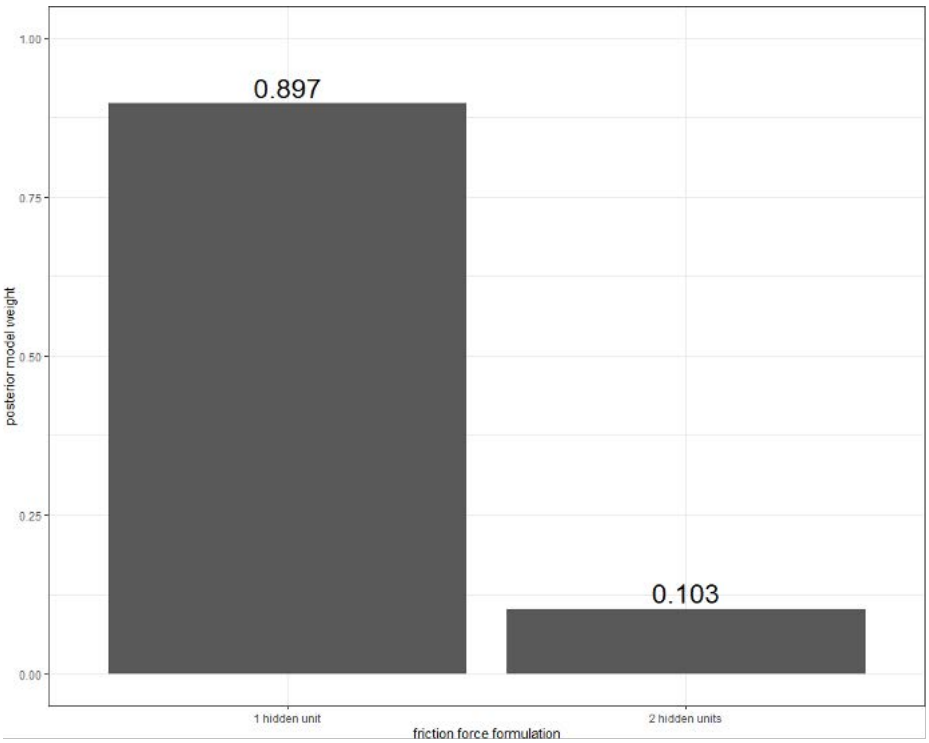


Figure 85. Posterior model weights for the two neural network contact force models.

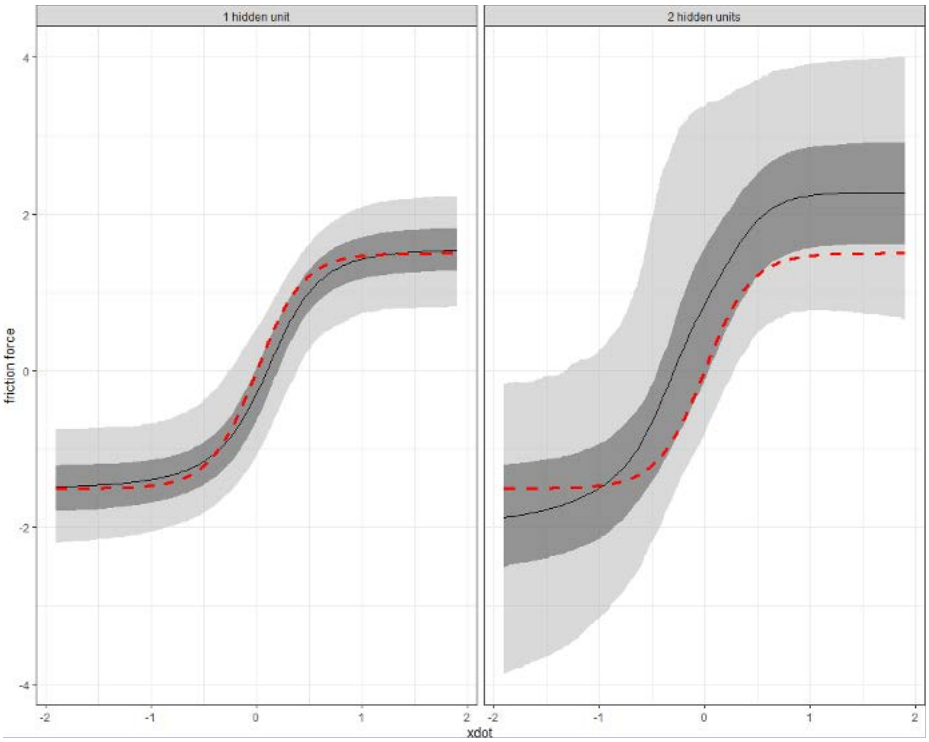


Figure 86. Posterior predictive distributions on the contact force with respect to velocity for the single hidden unit and two hidden unit neural network models.