BAYESIAN NETWORK REPRESENTING SYSTEM DYNAMICS IN RISK ANALYSIS OF NUCLEAR SYSTEMS

By

Athi Varuttamaseni

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering and Radiological Sciences)
in The University of Michigan
2011

Doctoral Committee:

Professor John C. Lee, Chair
Professor Xiuli Chao
Professor William R. Martin
Assistant Research Scientist Volkan Seker
Robert W. Youngblood, Idaho National Laboratory

To my parents

ACKNOWLEDGMENTS

I would like to thank members of my doctoral committee all of whom have helped me at various times during my research. I would especially like to express my gratitude to Professor John C. Lee for his supervision, guidance, and assistance on various matters that came up during my work. I would also like to give a special acknowledgement to Timothy Drzewiecki for his extensive help with the RELAP5 code. Finally, I would like to thank my parents for their unwavering support and understanding throughout my endeavors.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER 1

INTRODUCTION

1.1 Risk Assessment in the Nuclear Industry

The safe operation of nuclear power plants (NPPs) requires a careful analysis of the consequences of all events that can lead to the release of radionuclides to the environment. The initiating events can range from the failure of key safety systems which may not be immediately detectable to breaks in the primary coolant pipes which can cause damage to the fuel rods within a short period after the accident. Safety analysis, performed both before the construction of the NPPs and during the operation whenever there is a change in the configuration of the plants, can give a quantitative estimate of the risk that accidents or unforeseen events that occur in the plant will lead to offsite consequences.

Risk is estimated as a combination of the frequency of the initiating event (for instance, an accident) and the consequence of that event. Risk analysis attempts to determine how the initiating event will progress in time, incorporating the information on the responses of plant personnel and plant equipment. Risk analysis is especially crucial in the nuclear industry since failures of the safety barriers in an NPP may lead to a large release of radiation to the general public. Even if the plant were shut down immediately after an accident, the decay heat generated in the core will still require the operation of the cooling system at an adequate capacity. This level of dependence on the proper function of plant equipment long after shutdown means that there are many possible

evolution paths that an initiating event may follow with time. Risk analysis would therefore require a careful prediction of these evolution paths that may result in the undesirable consequences.

One of the earliest comprehensive probabilistic risk assessment (PRA) studies of nuclear power plants is WASH-1400 [NRC75]. This study focused on two representative power plants: Surry Power Station Unit 1 (a pressurized water reactor), and Peach Bottom Atomic Power Station Unit 2 (a boiling water reactor). The methodology employed in the study involves combining the fault tree and event tree structures. Fault trees are used to determine the combination of equipment faults that will lead to the failure of plant systems while the event trees are used to determine how different paths in accident evolution can lead to releases of radioactivity. Both the accident sequence and amount and mode of radioactivity releases are grouped into categories to make the analysis manageable. The WASH-1400 is also notable in its use of probabilistic density functions (PDFs) to account for uncertainties of event probabilities [Lee11]. The Zion and Indian Point probabilistic safety assessment studies [Hay99] were conducted around 1982. These studies used the impact vector methodology to simplify the PRA analysis and focused on evaluating the safety of these two plants considering their location near the urban centers.

The next major NRC-sponsored PRA study of nuclear power plants occurred a little over a decade after WASH 1400 and is commonly referred to as NUREG-1150 [NRC90b]. This study looks at three additional NPPs in addition to the two studied in WASH-1400. NUREG-1150 examines the accidents starting from any event that can lead to plant damage and follows the evolution until the release of radionuclides to the public

and evaluates the consequences of such a release in terms of fatalities. The accident progression event tree (APET) is constructed based on both expert solicitation and simulation codes (which in turn are validated through experiments and operation logs). Accident progression analysis generates a large number of possible accident pathways and is simplified by grouping of the sequence into bins based on the similarity of containment failure mode, probability, and time [Hak08]. The consequence analysis includes the use of radionuclide transport model to predict the spread of radioactivity after containment failure. The final risk integration gives the overall risk of the NPP to the public and is based on both the probability of the accident or malfunction and the consequences associated with them [Lee11].

1.2 Limitation of Static Risk Analysis

One of the limitations of PRA conducted using the static fault tree (FT) and event tree (ET) is that the dynamic response of the plant over time from an initial perturbation is not explicitly represented [Buc08]. The plant dynamics, arising from the interaction of different plant components and from the interaction between the operator and the plant control equipment, cannot be easily predicted during the construction of the static PRA models. The FT/ET approach usually assumes that the accident scenario can be represented as a static grouping of equipment failures or operator failures [Ald92]. Although during the construction of the ET, plant simulations are used to determine the different accident progression paths that the plant may take, it cannot easily account for how parameters such as the time delays in the actuation of safety systems will affect the accident progression. Furthermore, the actions of the automatic control systems depend

on both the time-dependent plant parameters and the corrective actions performed by the operator, whose state of knowledge depends on the path of the accident evolution [Kun08, Lab05]. These dependencies make it difficult to predict ahead of time the most likely set of paths in which the accident will evolve.

All the possible dependencies of the accident progression mentioned above means that an ideal risk analysis model should account for:

(1) the current status of all the relevant plant components,

(2) the current status of the plant variables (e.g., pressure and temperature),

(3) the current state of knowledge of the operator ,

(4) the evolution history during the transient of the plant variables, and

(5) the response model of both the automatic control systems and the operators.

The conventional FT/ET approach can account for the plant component status, the anticipated response of the control systems, and to some extent, the anticipated change in the system variables. However, it ignores how observations made by the operators during the transient can impact the action of the operators. Also, since the response of the plant variables is assumed during the construction of the model, it is based on the most likely plant configuration and configuration changes that the modeler thinks will occur during the accident. Any deviation from these assumptions can lead to changes in the plant response that can render the model invalid [Kop05].

The limitations mentioned above were recognized as far back as in the 1980s, when the concept of dynamic PRA was first developed in detail [Aco92]. However, many of the dynamic PRA techniques that have been proposed have found limited applications in the risk analysis of large, complex systems. For any system, the evolution of the

accident can take many different paths, each leading to different undesirable consequences (e.g., core damage) at different times. For large systems, this leads to situations where the complexity of the model grows exponentially with time. Today, detailed dynamic PRA studies are usually done for a smaller subsystems in situations where either the interactions between the process variable and the control systems are critically important, or in cases where human behavior needs to be explicitly modeled.

Over the past few years, there have been recognitions that even though static PRA may be insufficient for certain instances where the interaction between plant processes and control systems is important, we often do not need the full power of dynamic PRA methodologies and their associated complications. Several authors have proposed the use of analysis techniques borrowed from the study of artificial intelligence such as the Petri net, artificial neural network, and Bayesian networks to bridge the gap between static and dynamic PRA [Mur02, Por10, Shi08]. Due to their ease of construction and use, graphical methods such as the Bayesian network are especially attractive, especially compared to dynamic PRA methodologies such as the continuous event tree where the construction of the model can be difficult and time consuming. In this dissertation, we will show an extension on the use of the Bayesian network in risk analysis to include the incorporation of the reactor transient code surrogate that can model the dynamics of the reactor during accidents. This linking of the Bayesian network to a fast-running code surrogate will allow risk analysis to be performed in such a way that plant dynamics are directly incorporated into the model. The technique can serve as a bridge between conventional static PRA techniques and a full blown dynamic PRA model.

1.3 Research Scope

The research presented in this dissertation will focus on developing an appropriate code surrogate that can accurately represent the behavior of the reactor during a loss-of-feedwater (LOFW) transient and linking it to the dynamic Bayesian network model representing the system. Using such a combination will allow the risk analysis to be performed in such a way as to directly account for the dynamics of the plant while maintaining the simplicity (as least for model construction) of static PRA. As mentioned in the previous section, although there exist various dynamic PRA methodologies that do explicitly use system dynamics in the risk analysis, the models using these techniques are either too difficult to construct for large systems or will take too long to solve using today's computing capabilities. The work presented here is not meant to replace either the static or dynamic PRA. Rather, it is one additional tool that will be available to risk analysts should situations arise where the application of this method will help reduce the burden of the analysis of the safety of dynamic systems.

The dissertation will introduce the use of Bayesian network in dynamic PRA over several chapters. Chapter 2 will cover the currently existing methods for dynamic PRA and a simple problem that is commonly used to benchmark new dynamic PRA methodology. It will also describe some previous uses of the Bayesian network in risk analysis. Chapter 3 will describe the theory of the Bayesian network and its extension to the time domain in the form of dynamic Bayesian network (DBN). It will also revisit the simple problem introduced in chapter 2 and shows how this problem can be modeled with the BN. Chapter 4 will introduce the alternating conditional expectation (ACE) technique, a nonparametric regression technique that we will use to develop simplified models (i.e.,

the code surrogate) to represent system dynamics during an accident. Chapters 5 and 6 describe the LOFW transient in the pressurized water reactor and a RELAP5 model that is used to study the behavior of the plant during this accident. Chapter 7 presents the results of the risk analysis during the LOFW transient performed with a DBN. The dissertation concludes with chapter 8 with a discussion and ideas for possible extension of the DBN model to include applications in both diagnostics and real time monitoring and updating of data.

CHAPTER 2

Dynamic Probabilistic Risk Analysis

2.1 Survey of Available Techniques

Dynamic PRA methodologies that are available today are divided into three broad classes: dynamic event tree, discrete event transition models, and Monte Carlo simulation. The concept of the dynamic event tree extends the static event tree by allowing for branching at arbitrary times during the transient. The branching can be based on both the deterministic changes of the plant process variables and the stochastic transitions (failures) of plant components. The discrete event transition models are based on the discretized state of the plant. The plant state is defined by plant parameters (since these parameters are usually continuous, discretization is required) and the state of the components. Generally, the transition rate between plant states are dependent on the paths that the plant takes to reach the current state. However, keeping track of such a history would require an enormous amount of memory and may be impractical. Therefore, a Markov assumption is usually used. Under this assumption, the probability of transition to another system state is taken to be only dependent on the current state and is independent of how the system reaches the current state.

The Monte Carlo simulation represents the plant evolution based on component state transition probabilities. Although the simulation technique can be used to analyze a large variety of dynamic problems with very little assumptions, the main drawback is that

a proper sampling technique is required so that rare events (which cover the majority of the accident scenarios in the NPP) are adequately sampled.

2.1.1 Dynamic Event Tree

The dynamic event tree (DET) technique traces the plant state as it evolves in time after an initiating event [Hak08]. Within a certain time interval, the plant state may change from either the stochastic change of the component state (for instance, from an equipment failure) or from the deterministic change of the system variable as governed by plant physics. These deterministic changes can be obtained either from a suitable plant simulation code or from sets of equations governing the physical parameters. For cases where the plant behavior follows the Markov assumption (no dependence on evolution history), the dynamic event tree technique requires little computer memory. However, for cases where the history dependence is important (for instance, cases where observations of the operators play a role in future behavior [Hol99]), storing all the pathways of the accident evolution can be memory intensive. The dynamic event tree technique is also not suitable for sensitivity analysis since any change in the rule set or failure probability will require reanalyzing the entire scenario.

The DET methodology has been applied to many problems. Acosta looks at the application in the analysis of the steam generator tube rupture accident [Aco93]. Amendola and Reina applied the idea to the DYLAM code [Coj93] which allows the analysis of small subsystems of the NPP [Ald92]. Labeau [Lab050] extends the idea of the DET to include cases where the branching can be induced by the occurrence of certain stimuli which can be specified during the model construction.

2.1.2 Monte Carlo Simulation

Direct simulation of the plant behavior involves sampling the possible branching of the plant state based on a specified probability distribution. Unlike the case of the dynamic event tree approach which determines the branching based on a pre-specified set of branching rules, the Monte Carlo simulation samples from a large number of possible branches. This method of sampling leads to one major disadvantage for the Monte Carlo simulation: the difficulty of adequately representing rare events. Appropriate biasing is required to ensure that the result represents all the paths that can lead to core damage during the time interval of interest. Another disadvantage of the Monte Carlo method is that since the final result is an overall probability of reaching a particular end state, it is difficult to gather information about any particular progression pathway. Like the dynamic event tree approach, perturbation of the failure probability usually requires running a completely new simulation.

The Monte Carlo simulation technique applied to the water tank demonstration problem is often used in benchmarking other PRA techniques [Mar96]. This problem will be discussed in more detail in section 2.2 and will be used as a benchmarking case for the DBN technique in chapter 3.

2.1.3 Discrete Transition Models

The discrete transition model discretizes the system state and component state into different bins [Bel06]. The system evolution itself is represented by the transition of this combined state between different bins. Under the Markov assumption, the transition rate

depends only on the current state, and possibly on the transient time, but not on the detail of how the plant reaches its current state.

Risk analysis using Markov chains for static systems defines the system state as the state in which the system (i.e., all the components) resides. However, for dynamic systems, we usually make a distinction between the component state and system state. Let $x = (x_1, x_2, ..., x_N)^T$ be the vector describing the state of the system variables x_k, $k \in \{1, 2, ..., N\}$. These system variables are plant parameters such as pressure, temperature, and water level that evolve deterministically with time, given a certain plant configuration. The time evolution of the system variables is described by

$$\frac{dx}{dt} = f_i(x), \tag{2.1}$$

where $f_i(x)$ is the function describing how the system vector *x* will evolve with time, given the component state *i*. Here, the component state refers to a vector describing the state of the plant equipment such as valve position, pump speed, or control rod position. At any given time *t*, the NPP can be described by the pair (x_t, i_t). Our goal is to find the conditional probability density $\pi(x, i, t \mid x_0, i_0, t_0)$, where (x_0, i_0, t_0) is the initial state of the plant. If we now assume that the system is Markovian, Eq. (2.1) will satisfy the differential Chapman-Kolmogorov equation:

$$\begin{aligned} \frac{\partial \pi(x, i, t \mid x_0, i_0, t_0)}{\partial t} + \nabla_x \bullet \left(f_i(x) \pi(x, i, t \mid x_0, i_0, t_0) \right) + \lambda_i(x) \pi(x, i, t \mid x_0, i_0, t_0) \\ - \sum_{j, j \neq i} p(j \to i \mid x) \pi(x, i, t \mid x_0, i_0, t_0) = 0. \end{aligned} \tag{2.2}$$

Note that the probability $p(j \to i \mid x)$ that the component state vector will change from state *j* to state *i* depends on the system state *x*. An example of this dependence is when the failure probability of a valve depends on the coolant temperature. The transition probability $\lambda_i(x)$ is the probability that the component state leaves state *i*, given the system state *x*:

$$\lambda_i(x) = \sum_{j,\, j \neq i} p(i \to j \mid x). \tag{2.3}$$

We should note that in the case where the component failure probability is independent of the system state, we can integrate Eq. (2.2) over all system state x_0. Defining the component transition probability integrated over system states

$$\pi(i,t) = \int \pi(x,i,t)dx, \tag{2.4}$$

Eq. (2.2) becomes

$$\frac{d\pi(i,t)}{dt} = -\langle \lambda_i \rangle \pi(i,t) + \sum_{j,\, j \neq i} \langle p(j \to i) \rangle \pi(j,t). \tag{2.5}$$

Equation (2.5) is the Markov equation commonly used for reliability analysis problems with failure rates that are independent of the deterministic system state [Dev92].

Suppose Eq. (2.1) has the solution $x(t) = g_i(t, x_0)$. Equation (2.2) can be converted to an integral equation [Dev92, Lab05]:

$$\begin{aligned} \pi(x,i,t) = {} & \int \pi(u,i,t_0)\,\delta\big(x - g_i(t,u)\big)\big(1 - F_i(t,u)\big)\,du \\ & + \frac{1}{\lambda_i(x)} \int_{t_0}^{t} \int \lambda_j(u)\pi(u,j,t-\tau) \frac{p(j \to i \mid u)}{\lambda_j(u)} \delta\big(x - g_i(\tau,u)\big)\,dF_i(\tau,u)\,du, \end{aligned} \tag{2.6}$$

where

$$F_i(t,u) = 1 - e^{-\int_{t_0}^{t} \lambda_i g(s,u)\,ds}. \tag{2.7}$$

The integral equation (2.6) can be solved using Monte Carlo methods [Ald92].

To avoid the need for working with continuous state space, we can discretize Eq. (6). We start by defining the cell-averaged probability as:

$$\pi_k(i,t) = \int_{D_k} \pi(x,i,t)dx, \tag{2.8}$$

where D_k is the partition of the state variable x: $x \in \bigcup_k D_k$. Defining the indicator function

$$H_k(x) = \begin{cases} 1, & \mathrm{x} \in \mathrm{D}_k \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

we can integrate Eq. (6) over the cell D_k and obtain

$$\begin{aligned} \pi_k(i,t) = & \int \pi(u,i,t_0) H_k\left(g_i(t,u)\right) \exp\left(-\int_{t_0}^{t} \lambda_i g_i(s,u)ds \right) \\ & + \sum_{j,j\neq i} \int du \int_{t_0}^{t} d\tau \pi(u,j,t-\tau) H_k\left(g_i(t,u)\right) p(j \to i \,|\, u) \exp\left(-\int_{t_0}^{t} \lambda_i g_i(s,u)ds \right). \end{aligned} \tag{2.10}$$

It is possible to replace the transition probability $p(j \to i \,|\, u)$ by $p\left(j \to i \,|< u >_l\right)$, where $< u >_l$ is the average value of the system state over the cell D_l . Furthermore, we can replace the indicator function H_k by its cell average. These approximations lead to an averaged form of Eq. (2.10) which forms the basis of the cell-to-cell mapping technique [Wan04].

The Monte Carlo method and the dynamic event tree approach to solving dynamic PRA problems are not exclusive of one another. A combination of these two methods called the MCDET (Monte Carlo Dynamic Event Tree) was proposed by Hofer and

described in [Hof10]. In MCDET, the continuous event tree is discretized by means of Monte Carlo simulations. Discretization here refers to the sampling of the transition time and output state from the continuous event tree. The simulation yields groupings of the sampled paths which can then be analyzed.

Recently, interest in the use of graphical techniques in dynamic PRA analysis has led to a new class of techniques [And09]. Dynamic reliability analysis methods that are based on an extension of the static method includes the dynamic fault tree (DFT). A DFT is essentially a conventional static fault tree with several new dynamic gates added [Shi08]. The new gates include the functional dependency (FDEP) gate, the warm and cold spare gates, the priority AND (PAND) gate, and the sequence enforcing (SEQ) gate. The FDEP gate sets the output to true if the trigger is true. The warm and cold spare gates are used for components that are in standby. The PAND gate is true only if the basic event is true in a predetermined order. An extension of the PAND gate to more than two basic events is the SEQ gate. Each of the gates in the DFT can be converted to an equivalent Markov model and solved via the standard methods such as the matrix equation and Monte Carlo simulations [Pou08].

2.2 The Water Tank Problem

The water tank problem and its variation are often used to benchmark new dynamic PRA methodologies [Mar96]. The problem is based on a water tank with a prescribed inlet and outlet flows. The controller controls both flows based on the current water level of the tank.

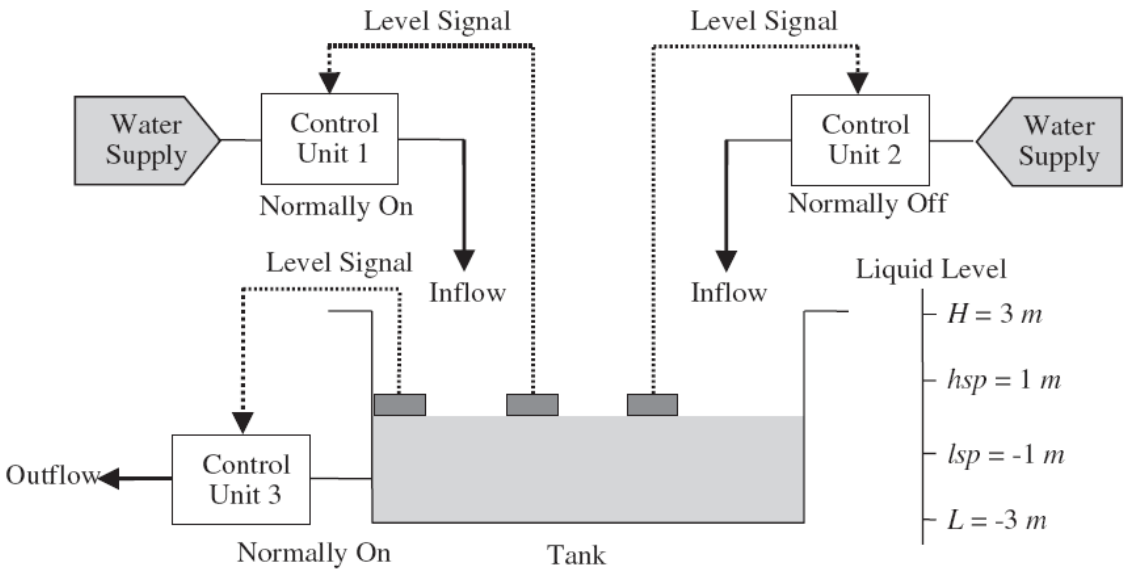

Figure 2.1. Water tank problem. Picture adapted from [Puc08].

Figure 2.1 shows the basic set up of the problem. We start from an initial value where the liquid level is at 0 m. Each of the three control units can be in one of four states shown in Table 1.1. We assume for simplicity that there is no repair and so if the controller fails either to state 3 or state 4, it will remain in that state forever (i.e., states 3 and 4 are absorbing states). Each of the three controllers will adjust its state (between state 1 and 2) based on the water level in the tank, with the goal of maintaining the water level between the lower set point *lsp* and the high set point *hsp*. This control behavior is summarized in Table 2.2.

Table 2.1. State space for each controller.

Controller State Identifier	Controller State Description
1	OFF
2	ON
3	Failed OFF
4	Failed ON

Table 2.2. Response of the controller to water level.

Water Level *h*	Controller 1	Controller 2	Controller 3
lsp < *h* < *hsp*	1	1	1
h < *lsp*	2	2	1
h > *hsp*	1	1	2

We will assume that the probability of failure into either states 3 or 4 is independent of the original state. The failure time is assumed to be exponentially distributed with the time constants of 219 hr, 175 hr, and 320 hr for controllers 1, 2, and 3, respectively. The pump connected to each of the controllers has the capacity to change the water level by 0.6 m/hr. For this problem, the differential equation governing the water level has a simple analytic solution:

$$h(t) = h(t_j) + Q(a_1 + a_2 - a_3)(t - t_j), \tag{2.11}$$

where

$$a_i = \begin{cases} 1, \text{ if } i \text{ is in state 1 or 3} \\ 0, \text{if } i \text{ is in state 2 or 4,} \end{cases} \tag{2.12}$$

and $t_j < t$ is the previous time step. The mission time is 1000 hr. The goal of the problem is to find the time dependent probability that the tank will be in either the dryout ($h < L$) or overflow ($h > H$) state.

In the problem described above, the failure rate is independent of the system variable (the water level). This means that classical static PRA methods such as ET/FT and Markov chain can be easily used to solve the problem. However, complications will arise if we allow for cases where the component failure rate depends on the component state or the process variable. In these latter two cases, using static PRA to analyze the problem will present more difficulty. Marseguerra [Mar96] discusses several variations to the water tank problem and how the Monte Carlo approach to dynamic PRA can be used to evaluate the probabilities of overflow and dryout. In chapter 3, we will solve the basic water tank problem with the dynamic Bayesian network.

CHAPTER 3

THE BAYESIAN NETWORK

The Bayesian network (BN) is used in many diverse areas of application ranging from medical diagnosis to speech recognition. In this chapter, we will describe the basics of a BN, how they are constructed, and how they can be solved. The theory that is available to support the analysis of the BN is very broad and most will not be needed for a successful application of the BN to certain specific problems. Nevertheless, we will briefly touch on the major developments and applications of the BN so that the versatility of this model can be appreciated.

The application of BN to reliability analysis is a relatively recent development. The static form of the network has been applied to study the risk associated with nuclear waste disposal [Lee06] and to study the impact of equipment maintenance on reliability [Web10]. Montani and Portinale [Mon08] developed a reliability tool called RADYBAN which uses the BN to solve dynamic fault tree problems. Boudali [Bou05] looks at the formulation of the continuous form of a temporal BN to calculate the time to failure of an equipment network. In this sense, the BN can be applied to problems that are traditionally analyzed using static fault trees. In their review paper, Langseth [Lan05] discusses how the BN can be used to incorporate the quality of opinions of different experts directly into a risk model.

A key feature of the BN that is appealing to many analysts is that the network itself can be constructed based on intuition that is developed through experience without the need for detailed knowledge of the analysis techniques. Although this feature is shared by other graphical techniques, BN is perhaps the most developed in terms of the underlying theoretical framework. In contrast, powerful techniques such as the Markov or semi-Markov models can give similar (sometimes more detailed) results compared to the BN, but with a greater effort required for their construction and analysis.

We will divide the presentation of BN theory into several sections. Section 3.1 discusses the basics of the static BN including the interpretation of the independency statement and the use of conditional probability function to quantify the network. Section 3.2 extends the static BN to include a time domain. There are several different ways of including time into the BN, each with their own advantages and disadvantages. Methodologies available to solve the BN is discussed in Section 3.3. Today, there are a number of commercial and open source software available for BN analysis. The most suitable method used to solve the network can depend on the size of the network, the type and amount of data available to populate the network, and the type of information we wish to obtain from the network. In Section 3.4, we will present an application of the BN to solving a dynamic fault tree problem using a continuous formulation of the network. Finally, we will end the chapter by applying the BN to analyze the water tank problem that we presented in Chapter 2.

3.1 Static Bayesian Network

A Bayesian network is a directed acyclic graph with nodes representing random variables, and directed arcs linking the nodes, representing conditional dependencies between different variables. It is convenient to classify the nodes as either a parent node or a child node. The parent node is a node that has one or more arc originating from it and pointing towards another node. A child node is a node with incoming arc from another node. A node with no parent is referred to as the root node and a node with no children is the leaf node. For instance, in Figure 3.1, nodes *A* and *B* are the parents of node *D* (this implies *D* is the child node for nodes *A* and *B*). Nodes *A* and *B* are root nodes while nodes *C* and *E* are leaf nodes.

Each parent node *N* in a BN is associated with the marginal probability distribution $P(N)$ and each child node *M* has an associated conditional probability $P(M|pa(M))$, where $pa(M)$ is a set of all the parents of *M*. Referring again to the BN in Figure 3.1, a complete description of the BN would include, in addition to the graph in Figure 3.1, specification of the probabilities $P(A)$,$P(B)$, $P(C)$, $P(C|A)$, $P(D|A,B)$, and $P(E|D)$.

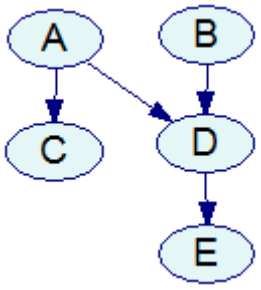

Figure 3.1. A representative Bayesian network.

A graphical representation of the BN like the one in Figure 3.1 shows the conditional independence among the random variables. Generally, a node in a BN is conditionally independent of its non-descendants given the parent nodes. (A node F_1 is a descendant of node F_2 if there is a directed path from F_2 to F_1.) In Figure 3.1, the node E is conditionally independent of nodes A and B given that the state of node D is known:

$$P(E \mid A, B, D) = P(E \mid D). \tag{3.1}$$

Conditional independence statements like the one in Eq. (3.1) can be used to simplify the joint probability of a system. For any random variables X_1, X_2,..,X_n we can factorize the joint probability, without any assumption, using the chain rule as:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, ..., X_{i-1}). \tag{3.2}$$

However, if we exploit the property of the BN that a variable is independent of its non-descendants given the parents, then Eq. (3.2) reduces to:

$$P\left(X_1, X_2, ..., X_n\right) = \prod_{i=1}^{n} P\left(X_i \mid pa\left(X_i\right)\right). \tag{3.3}$$

The key property of the BN technique is that it factorizes the joint probability distribution. If we have a set of N random variables $\{X_1, .., X_N\}$, all the information that we can ever know about these variables can be obtained by knowing the joint distribution function of the variables. However, specifying the joint distribution function is a complicated task. For a discrete random variable, suppose we assume that each of the X_i in the set above can be in any one of m possible states. This means that to completely specify the joint distribution, we need to provide $m^N - 1$ numbers. There are several problems associated with this. The most obvious problem is that it will be hard to use

this information, especially if *m* and *N* are both large. Even with today's computer, the processing and storage of this amount of information will be slow and expensive. Oftentimes, we do not need all the information that is contained in the joint distribution. However, it is difficult to select the information that we do need without at least some preliminary analysis of the full distribution.

The second problem is the difficulty of obtaining the information needed to specify the $m^N - 1$ probabilities. Information from expert elicitation usually does not come in the form of probabilities, and even if they do, most likely not in the form that can be directly entered into the joint distribution. An alternative is to automatically update the $m^N - 1$ probabilities with data. However, it will take a lot of data, which may not be available, to update the probabilities enough times that the resulting joint distribution accurately represents the true distribution.

A BN allows the joint probability distribution to be factored by exploiting the conditional independence relations among the different variables. For instance, consider the BN shown in Figure 3.1. For simplicity, we will assume that each of the 5 nodes in the network can be in one of two states: TRUE or FALSE. If we were to specify the joint conditional distribution $P(A=i, B=j, C=k, D=l, E=m)$, we would need to specify 2^5-1=31 probabilities. However, using the conditional independence assumption of Figure 3.1, we can write:

$$P(A,B,C,D,E) = P(A)P(B)P(C \mid A)P(D \mid A,B)P(E \mid D). \tag{3.4}$$

The number of probabilities needed to specify $P(A)$ is one: we need to only specify $P(A = \text{TRUE}) = 1 - P(A = \text{FALSE})$. For the term $P(C \mid A)$, we only need to specify two

probabilities: $P(C = \text{TRUE} \mid A = \text{FALSE})$ and $P(C = \text{TRUE} \mid A = \text{TRUE})$. Using a similar reasoning for the other terms, we see that the total number of probabilities that we need to specify to determine the RHS of Eq. (3.4) is 1+1+2+4+2=10. Therefore, we see that the assumption of conditional independence can be used to simplify the network. More generally, for a BN with *n* nodes with *k* parents or less, the complete specification of the joint distribution requires an order of $(2^k)n$ numbers. The unfactorized form of the joint probability requires an order of 2^n numbers.

3.2 Dynamic Bayesian Network

A dynamic Bayesian network (DBN) is a BN that incorporates nodes that can change with time. The simplest way to account for the time dependence is to divide the time line into different time "slices" and assign one complete static BN to each slice. The time dependence then is represented by an arc from the lower numbered slice to a higher numbered one. If the nodes for slice at time *t* is dependent only on nodes from slice *t*-1, then we say that the DBN is a two-slice DBN. For a general case, the dependencies may extend to any time slice in the past.

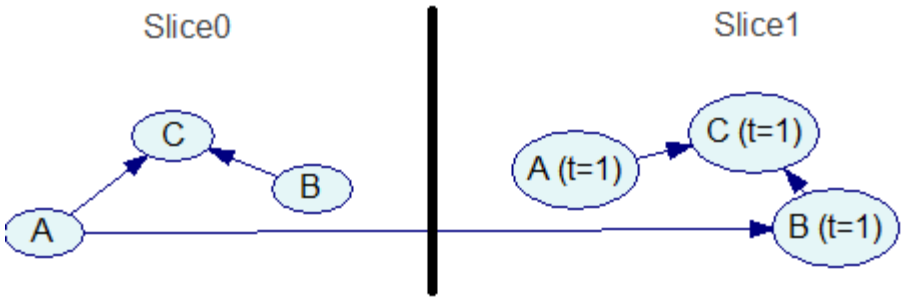

Figure 3.2 A simple dynamic BN expanded over two time slices.

A simple DBN is shown in Figure 3.2. For any given slice, node *C* is completely determined by its parents (nodes *A* and *B*). The parent node *B* at any given slice depends on the value of node *A* at the preceding slice. The arc linking *A* at slice 0 to *B* at slice 1 is referred to as an interslice arc. Interslice arcs indicate the dependence of the nodes that are time dependent.

The methods that are available to solve a static BN can be applied to dynamic DBN. The interslice arcs are treated as any regular arcs. Nodes representing the same variables but located in a different time slice are considered to be distinct for the algorithm.

The description of the DBN above applies for what is called an instant-based approach to temporal analysis. Another approach of introducing time into the BN is called the interval-based approach. In this approach, the random variables themselves represent events that are time dependent. The time line is usually divided into different disjoint intervals with the node (representing the random variable) defined according to the intervals in which it is located. The DBN models used for our work are all instant-based.

3.3 Continuous Bayesian Network

So far, we have limited our discussion to BNs with nodes that represent discrete random variables. Generally, a BN has no limitation on the type of node it can have. A BN that contains both discrete nodes and continuous nodes are known as mixed or hybrid BN. Most of the inference algorithms that have been developed over the last few decades only apply to a discrete BN or to continuous BN with certain specific distributions.

Nevertheless, a continuous BN has several advantages over its counterpart. First, there is no need to discretize the sample space of the variables. This means that common problems that are associated with discrete BNs such as the state space explosion do not exist with the continuous version. Second, since many applications of the BN involve the use of continuous variables, the continuous form of the network is the natural choice to represent these variables. In these instances, the only real reason to discretize the variables is to have access to a greater selection of inference algorithms that can be used. Finally, the solution of continuous BNs can be expressed as integral equations which may be solved by various standard techniques. The ability to write an analytic solution to a BN problem may provide insights into the solution which are more difficult to obtain if we limit to only the numerical relations.

In this section, we will present a simple illustration on the use of the continuous BN. The problem is to use the continuous BN formalism to analyze the AND gate in a fault tree. The solution technique is based on the work by Boudali [Bou05] in which he presents the conversion of different dynamic fault tree gates to the BN equivalent. The simple BN model representing an AND gate is shown in Figure 3.3 below.

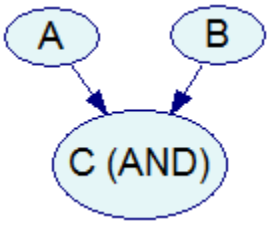

Figure 3.3. The BN representation of an AND gate.

The variables *A* and *B* represents to components, each of which can be in either the operational (FALSE) or failed (TRUE) state. The node *C* is TRUE only if *A* and *B* are

both TRUE, otherwise it is FALSE. Let nodes *A* and *B* have the time-to-failure distributions $f_A(t_A)$ and $f_B(t_B)$, respectively. The function $f_A(t_A)$ is the probability density function for component *A* to fail in the neighborhood of time t_A. A similar interpretation applies to $f_B(t_B)$. Next, we need to find the conditional time-to-failure distribution for the node *C*, given the states of nodes *A* and *B*. We note that if node *A* enters a failed state before node *B*, then the time to failure of the node *C* will be equal to the time-to-failure of node *B*. Conversely, if node *B* enters the failed state before node *A*, then the time to failure of node *C* will be the time to failure of node *A*. These two statements suggest that the conditional time-to-failure distribution for the node *C* will be:

$$f_{C|A,B}(t_C \mid t_A, t_B) = u(t_B - t_A)\delta(t_C - t_B) + u(t_A - t_B)\delta(t_C - t_A), \tag{3.5}$$

where $u(x)$ and $\delta(x)$ are the unit step function and the Dirac delta function, respectively. The first term on the RHS is nonzero only when *A* fails before *B* while the second term is only nonzero when *A* fails after *B*. The delta function selects the failure time for *C* based on the component that fails last. If we now assume that components *A* and *B* are independent, we can form the joint time-to-failure density function as:

$$f_{ABC}(a,b,c) = f_{C|A,B}(c \mid a,b)\, f_A(a)\, f_B(b). \tag{3.6}$$

Armed with the joint distribution shown in Eq. (3.5), we can calculate the density function for gate *C*. This can be done by marginalizing (integrating) Eq. (3.5) with respect to both *a* and *b*:

$$\begin{aligned} f_C(c) &= \int_0^\infty \int_0^\infty f_{C|A,B}(c \mid a,b)\, f_A(a)\, f_B(b)\, da\, db \\ &= f_A(c)F_A(c) + f_B(c)F_B(c) \\ &= \frac{d}{dc}\left[F_A(c)F_B(c)\right], \end{aligned} \tag{3.7}$$

where $F_A(x)$ and $F_B(x)$ are the cumulative time-to-failure distribution for components *A* and *B*, respectively. Let us compare this solution to the result from the Markov chain analysis. To do this, we will first need to assume that the time-to-failure for both components *A* and *B* are exponentially distributed:

$$F_A(a) = 1 - e^{-\lambda_A a}, \tag{3.8}$$

$$F_B(b) = 1 - e^{-\lambda_B b}, \tag{3.9}$$

where λ_i is the failure rate of component *i*. Substituting Eqs. (3.8) and (3.9) into Eq.(3.7), we obtain the cumulative time-to-failure distribution of the AND gate:

$$F_C(t) = 1 - e^{-\lambda_A t} - e^{-\lambda_B t} + e^{-(\lambda_A + \lambda_B)t}. \tag{3.10}$$

We note that Eq. (3.10) agrees with what we would expect if we approach the problem directly using a Markov chain approach.

A similar approach can be used to derive the time-to-failure distribution for other gate types of both the static and dynamic fault trees. For cases involving more than two states, the manipulation of the associated equations becomes much more complicated. For large real world problems, the continuous BNs are usually either discretized and converted to the static BN or solved via Monte Carlo simulations.

3.4 Application of Bayesian Network to the Water Tank Problem

We now consider the water tank problem described in Chapter 2. The objective of this exercise is to calculate the probability of overflow and dryout as a function of time. Figure 3.2 shows the BN associated with this problem. The dependence of the variables vary over two time slices. The states of the three control units determine the changes in water level as governed by Eq. (2.11). The water level in the tank at any time step is the

level in the previous step plus any changes based on the configuration of the control units. In the network, this relation is presented by one interslice arc from each of the control unit in the previous slice and another interslice arc from the water level in the previous time step. Since the control units themselves have a constant failure probability, the state of these units at any one time slice depends on the state at the previous slice. The probabilities associated with these interslice arcs are the stochastic failure probabilities of the control units over the time interval represented by the slice. In addition to the possibility of failure, the control units also respond to the water level. These response are represented by the arcs from the water level node to the control units node. However, unlike the arcs representing the control failure, the control arcs are deterministic.

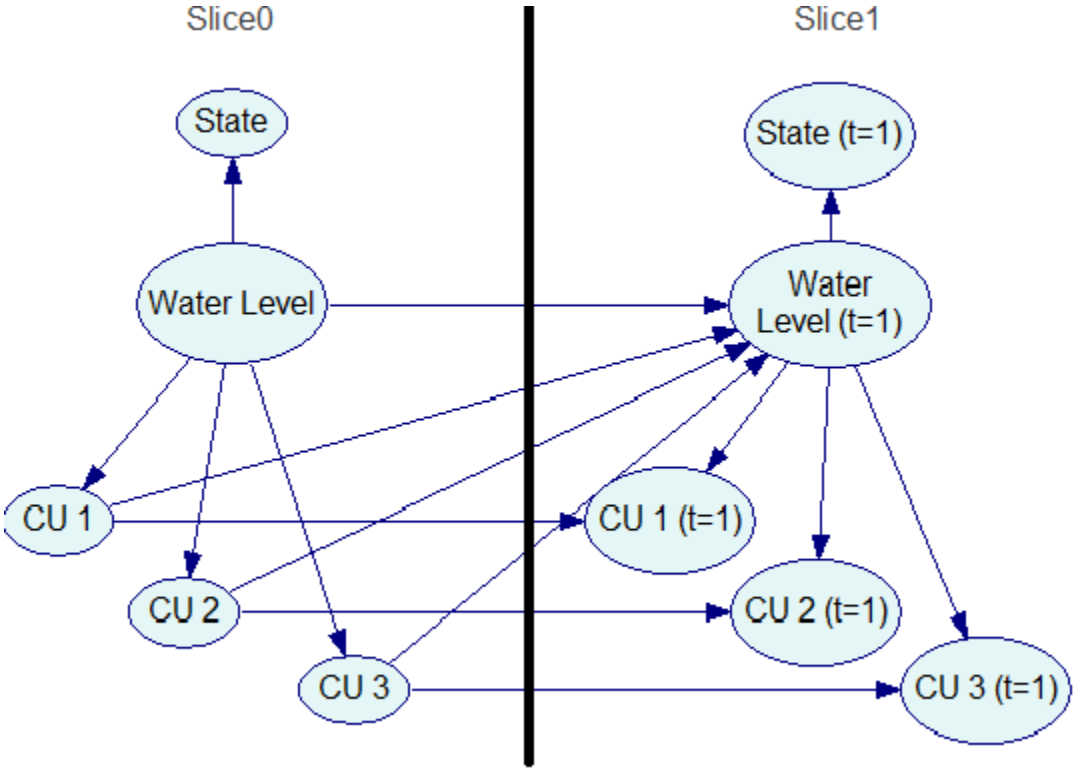

Figure 3.4. A DBN representing the control functions of the water tank problem.

The network in Figure 3.4 can be solved by starting at time 0 and solve the BN problem at time 0 with a prescribed initial condition. The result of this analysis will give the values to be used in the calculation of the network in the second slice. This process is repeated over all the time slices. The probability density for the water level in slice 1 can be written as:

$$P(W\#) = \sum_{W,CU1,CU2,CU3} P(W\#|W,CU1,CU2,CU3)P(W) \\ \text{x } P(CU1)P(CU2)P(CU3), \tag{3.11}$$

where the "#" symbol denotes the value in slice 1. At each time slice, the conditional probability function for the control unit *i* can be expressed in terms of the corresponding probability of the previous time slice:

$$P(CUi\#) = \sum_{CUi,W} P(CUi\#|CUi,W)P(W). \tag{3.12}$$

The joint probability density can be summed over all the ancestor nodes:

$$P(W\#) = \sum_{W,CU1,CU2,CU3} P(W\#|W,CU1,CU2,CU3)P(W) \\ \text{x } P(CU1)P(CU2)P(CU3), \tag{3.13}$$

The conditional probability function $P(CUi\#|CUi,W\#)$ that appears in Equation (3.13) is specified by the behavior of the control units as shown in Table 2.2. For instance, the control probability for control unit 1 is shown in Table 3.1. Here, we are assuming that each time slice represents a time step of Δ. The transition probabilities include both the effect of stochastic failures and deterministic controls. If the unit is already in any of the two failed states, we assume that it remains in that state (no repair).

At any particular point in time (at any particular slice), the analysis yields a fixed probability for the tank to be at a certain level. These levels are then used to determine whether or not the tank overflows or dries out. The probabilities at each time slice are

shown in Figures 3.5 and 3.6. The data from the Monte Carlo simulation are taken from [Mar96] and are generated from 10^5 simulations. The agreement between the two techniques is very good.

Table 3.1. Conditional probability function for control unit 1.

State at t		State at $t+\Delta$			
W	CU 1	XU1#			
		ON	OFF	FAILED ON	FAILED OFF
$hsp < W < H$	ON	0	1-2 $\lambda_1\Delta$	$\lambda_1\Delta$	$\lambda_1\Delta$
$hsp \le W \le lsp$	ON	0	1-2 $\lambda_1\Delta$	$\lambda_1\Delta$	$\lambda_1\Delta$
$L < W < lsp$	ON	1-2 $\lambda_1\Delta$	0	$\lambda_1\Delta$	$\lambda_1\Delta$
$hsp < W < H$	OFF	0	1-2 $\lambda_1\Delta$	$\lambda_1\Delta$	$\lambda_1\Delta$
$hsp \le W \le lsp$	OFF	0	1-2 $\lambda_1\Delta$	$\lambda_1\Delta$	$\lambda_1\Delta$
$L < W < lsp$	OFF	1-2 $\lambda_1\Delta$	0	$\lambda_1\Delta$	$\lambda_1\Delta$
any	FAILED OFF	0	0	0	1
any	FAILED ON	0	0	1	0

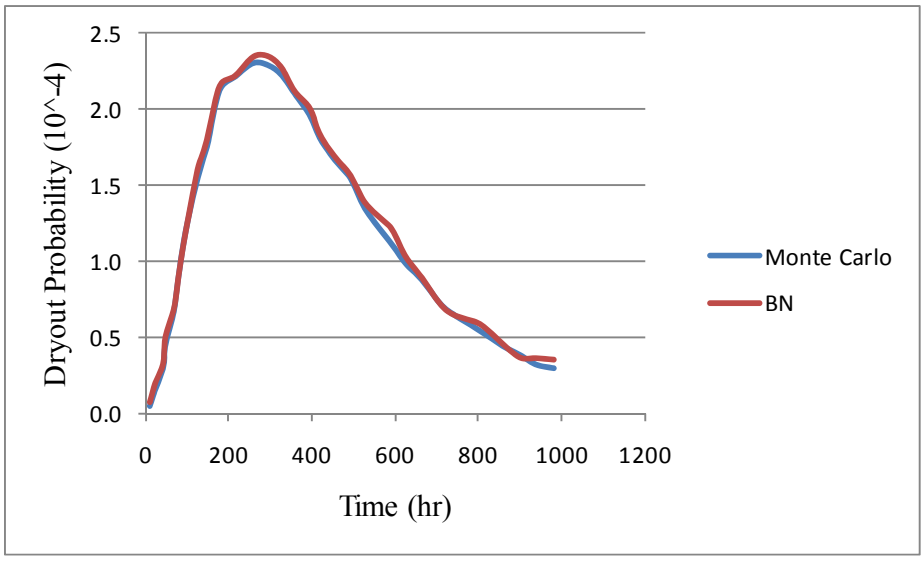

Figure 3.5. Dryout probability calculated using BN and Monte Carlo. The Monte Carlo result is from [Mar96].

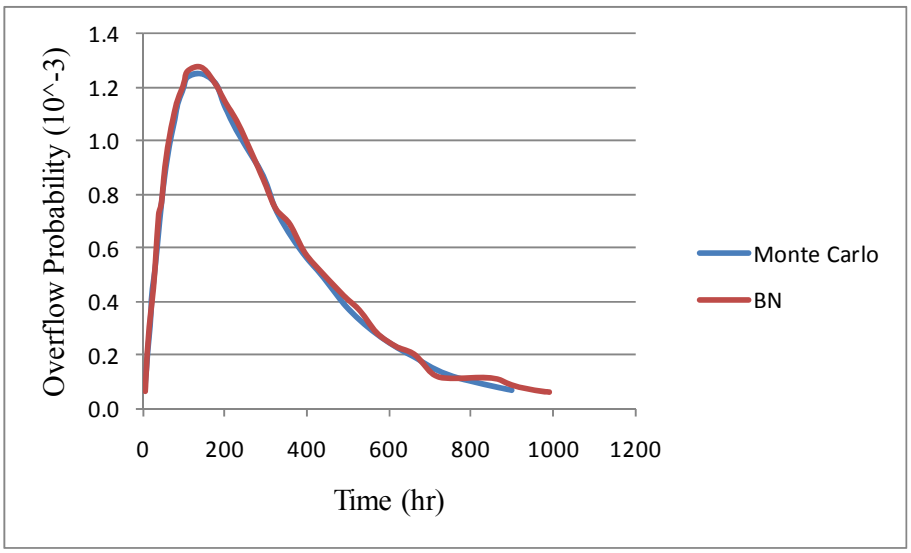

Figure 3.6. Overflow probability calculated using BN and Monte Carlo. The Monte Carlo result is from [Mar96].

CHAPTER 4

Alternating Conditional Expectation

In this chapter, we will introduce the use of the alternating conditional expectation (ACE) to construct a surrogate for any transient analysis code. ACE is a nonparametric, nonlinear multivariate regression technique that determines the optimal functional transformation of both the dependent variable and a set of independent variables such that a linear relationship between the transformed dependent and independent variables is maximized. However, before we discuss ACE in more detail, we will start explaining in Section 4.1 why a code surrogate is needed in our work. Section 4.2 will present the ACE algorithm and some of the physical interpretations that can be associated with the technique. We will also give a brief outline of the algorithm that is used to implement ACE. The final section of this chapter will apply the ACE technique to a simple function so that the benefit of ACE will become clearer.

4.1 Need for a Code Surrogate

Generally, the behavior of a reactor during accident-induced transients depends on the configuration of the plant components which themselves will vary with time. Advanced reactor analysis codes often will have the ability to automatically adjust the state of the components in response to a change in the reactor state variable. For instance, the auxiliary feedwater pumps can be set to automatically turn on when the water level in a

steam generator drops below a limit. This capability of these analysis codes to model automatic control means that it is easier for the analyst to focus on specific components important to the study and let the code adjusts the state of the other components. However, in risk analysis, we are interested in finding the overall probability of core damage, given any possible configuration changes of the reactor. This means that we need to specify probabilities of having a component transitioning to different states (e.g., a valve changing its state from OFF to ON) given the state of the plant. To perform this type of analysis, the analysis code needs to be run at all the possible (or at least most likely) combinations of plant component states and then determine how the transient behavior is affected. Considering that a detailed model of a reactor will include many components with many states, the number of all possible combinations is quite large. The run times for these analysis codes for a given case can vary from 10 minutes to several hours. It would take an impractical amount of time to use these codes to study all the possible variations of component states and determine the impact on reactor safety.

The idea behind a code surrogate is that we can replace the detailed but slow running analysis code with a somewhat less detailed but much faster representation. A surrogate is basically a function that specifies a deterministic relationship between a dependent variable of interest and a set of independent variables. For instance, we may be interested in a function that expresses the peak clad temperature in terms of coolant flow rate, pressure, and temperature. This surrogate will then be used instead of the simulation code if we want to know how the clad temperature changes as a function of variations in the coolant flow rate, pressure, and temperature.

A given surrogate will most likely have a range of applicability. It is important that if we use values of the independent variables that are far outside the range that we use to construct the surrogate, then the surrogate be checked against the original code. The work presented in this dissertation uses several layers of surrogates. In the example above, we pick the flow rate, pressure, and temperature as the independent variables. However, these are not variables over which we have direct control. Therefore, these variables are in turn expressed as a function of other independent variables. The choice of independent variables to use for a given surrogate is made mostly from experience or judgments made from understanding of the basic physics. In this regard, the accuracy of a particular surrogate model will vary from case to case.

In the next section, we will present the ACE algorithm which we use to build the various surrogates used in this work. We will defer the discussion of the actual surrogate construction until Chapter 6.

4.2 Overview of ACE

In linear regression analysis over *p* independent variables, we are interested in finding the coefficients $a_0, a_1, ..., a_p$ such that the equation

$$Y = a_0 + \sum_{i=1}^{p} a_i X_i + \varepsilon \tag{4.1}$$

holds. Here, ε is the error, Y is the dependent variable and X_i's are the independent variables. Linear regression works well if the relationship between the independent and dependent variables are known to be linear. However, in a more general case, linear regression may give erroneous results. This leads to the development on nonparametric

regression techniques which do not assume the functional form of the relationship. These nonparametric techniques are divided into two broad classes: those that perform some kind of transformation on the dependent variable, and those that do not. The ACE model falls into the former class. The ACE algorithm is first proposed by Breiman and Friedman [Bre85] as a way to estimate the optimal transformations of both the dependent and independent variables. By optimal transformation, we mean a transformation that reduces the mean squared error. A different criterion for optimality will lead to a different regression algorithm.

The ACE model expresses the transformed function θ of the dependent variable Y in terms of the transformed functions ϕ_i of the independent variables X_i, $i \in \{1, 2, .., p\}$:

$$\theta(Y) = \sum_{j=1}^{p} \phi_j(X_j) + \varepsilon, \tag{4.2}$$

where the noise ε has a zero mean and is independent of all the X_i's. The ACE algorithm aims to minimize the squared error

$$er(\theta, \phi_1, ..., \phi_p)^2 = E\left(\theta(Y) - \sum_{j=1}^{p} \phi_j(X_j) \right)^2. \tag{4.3}$$

For a given value of θ and ϕ_j, $j \neq i$, the value of ϕ_i that minimizes Eq. (4.3) (i.e., the solution of the minimization problem) is:

$$\phi_i(X_i) = E\left(\theta(Y) - \sum_{j, j \neq i}^{p} \phi_j(X_j) \mid X_i \right). \tag{4.4}$$

Likewise, for a fixed ϕ_j, the θ that minimizes Eq. (4.3) is:

$$\theta(Y) = \frac{E\left(\sum_{i=1}^{p} \phi_i(X_i) \mid Y\right)}{\left\| E\left(\sum_{i=1}^{p} \phi_i(X_i) \mid Y\right) \right\|}, \tag{4.5}$$

where $\|(\bullet)\|$ is the square-norm. The name "alternative conditional expectation" comes from the fact that the ACE algorithm alternates between solving the minimization problem with θ fixed [Eq. (4.4)] and keeping ϕ_j fixed [Eq. (4.5)]. The discussion above can be cast into an algorithmic form shown below [Bre85]:

(i) Initialize: set $\theta(Y) = \dfrac{Y - E[Y]}{\sqrt{\text{var}[Y]}}$

(ii) Obtain new ϕ: $\phi_k(X_k) = E\left(\theta(Y) - \sum_{i=1, i\neq k}^{p} \phi_i(X_i) \mid X_k\right)$

(iii) Obtain new θ: $\theta(Y) = \dfrac{E\left(\sum_{i=1}^{p} \phi_i(X_i) \mid Y\right)}{\sqrt{\text{var}\left[E\left(\sum_{i=1}^{p} \phi_i(X_i) \mid Y\right)\right]}}$

(iv) Alternate between (ii) and (iii) until $E\left(\theta(Y) - \phi(X)\right)^2$ does not change.

4.3 Data Smoothing

Data smoothing is performed when we are interested in finding the behavior of the dependent variable Y as a function of a set of N independent variables $X_i, i \in \{1,..,N\}$. The term "smoothing" is used since the output of a smoother generally has less fluctuation (i.e., smoother) than the original variable. In the derivation of the ACE algorithm in Section 4.2, we used the conditional expectation value of the transformed function given the original variable to perform the iteration. However, this works only if

we know the joint distribution functions of these variables. Without the distribution functions, we will not be able to calculate the conditional expectation and hence use ACE directly. This is where a smoother comes into play. The results of the smoothing process is used to replace the conditional expectations in the ACE algorithm. Note that since the smoothing process itself is nonparametric, its use in the ACE algorithm does not remove the nonparametric nature of the algorithm.

To formalize the notion of data smoothing, we will follow the development in [Bre85]. Suppose we have *N* sets of *p* independent variables. Mathematically, we can define this data set to be the set $\{X_1,...,X_N\}$ whose elements X_i is a point in the *p*-dimensional space. For this data set, define $F(X)$ to be a space of all real-valued functions *Y* whose domain is our data set $\{X_1,...,X_N\}$. We can think of $F(X)$ as a space of all the possible dependent variables. Also define $F(x_j), j \in \{1,..,p\}$ to be a space of all real-valued functions defined on the set $\{X_{1j},...,X_{Nj}\}$. We can think of $F(x_j)$ as a space of all possible functions of the independent variable x_j. The data smooth *S* of *X* on x_j is a mapping $S : F(X) \to F(X_j)$ that is defined for every data set. For every *Y* in $F(X)$, we write the corresponding element in $F(x_j)$ as $S(Y \mid X_j)$ and its values as $S(Y \mid X_{kj})$. Note that with this definition of smoothing, a linear regression that fits a line through a set of data point can be considered a smoothing technique. However, since linear regression assumes a linear relationship, it is not usually considered to be a real smoother.

An example of a simple smoothing technique is a running-mean smoother. Suppose we have a set of *N* data points of the form $(x_i, y_i), i \in \{1,..,N\}$ where x_i is the independent variable and y_i is the dependent variable. For any given value of x_i, we can

define a set $N^S(x_i)$ that contains *k* points closest to the left of x_i and *k* points closest to the right of x_i. A running mean smoothing function is then defined as:

$$S(x_i) = \underset{j \in N^S(x_i)}{Ave}(y_j). \tag{4.6}$$

The set $N^S(x_i)$ is referred to as the symmetric nearest neighborhood. What we are doing here is essentially averaging the values of the dependent variables within a certain neighborhood. The averaging process will smooth out any variations in the dependent variable. A big disadvantage of this technique is that in cases where the values of the outliers are very different from the values at the interior points, we will lose the information on the behaviors of the dependent variables near those outliers. To alleviate this problem, the running mean smoothing technique can be generalized to use other forms of neighborhood selection (including non-symmetric ones).

The choice of the smoothing technique to use determines the output of the ACE algorithm. In this dissertation, we will use the technique originally used in [Bre85] known as a "supersmoother" which basically performs a linear fit within a neighborhood of the point of interest. The size of the neighborhood (i.e., the window width) is allowed to vary, being governed the median of the neighborhood [Fri82].

We will conclude this section by noting that the ACE technique has only been proven to converge if the smoother used in the algorithm is linear. (A linear smoother satisfies $S(ay_1 + by_2 \mid x) = aS(y_1 \mid x) + bS(y_2 \mid x)$ for any constants *a* and *b*.) Nevertheless, the ACE algorithm has been successfully employed in many applications even with a more general choice of smoothers [Bre85].

4.4 A Simple Demonstration of ACE

In this section, we will demonstrate the use of the ACE technique to analyze a data set that is synthetically generated from a known functional relationship. This example is adapted from [Wan04]. Consider the equation:

$$e^{Y} = 4 + \sin(4X_1) + |X_2| + X_3^2 + X_4^3 + X_5 + 0.1\varepsilon. \tag{4.7}$$

The variables $X_i, i \in \{1,2,3,4,5\}$ are assumed to be independent with a uniform distribution between -1 and +1. The noise ε is independent of X_i and is assumed to be normally distributed with a zero mean and unit variance. We now sample 500 data points from this distribution to generate a set of independent variables and their associated dependent variable *Y*. With the data set $(X_1, X_2, ..., X_5, Y)$ generated, we wish to apply the ACE algorithm to try to recover the functional dependence of *Y* on the X_i's and compare the results to Eq. (4.7). Of course in practice, we will only have the 500 sample points of the set $(X_1, X_2, ..., X_5, Y)$ without knowing the relationship between the variables.

Faced with an unknown relation between the dependent variable *Y* and the independent variables $X_i, i \in \{1,2,3,4,5\}$, one naïve approach may be to look at the scatter plot of *Y* versus the individual X_i with the hope of guessing a functional form of the dependence. Such a plot is shown in Figure 4.1. Looking at the plots, we may perhaps guess that *Y* may be a sinusoidal function of X_1 while the dependence on the other variables may be linear.

If we now apply the ACE algorithm to the data set, we arrive at the relations shown in Figure 4.2. As can be seen from the figure, the ACE technique is able to recover the functional form of the dependent variables: sinusoidal function for X_1,

absolute value function for X_2, quadratic function for X_3, cubic function for X_4, and a linear function for X_5. The dependent variable *Y* is transformed to an exponential function, as expected from the relations in Eq. (4.7).

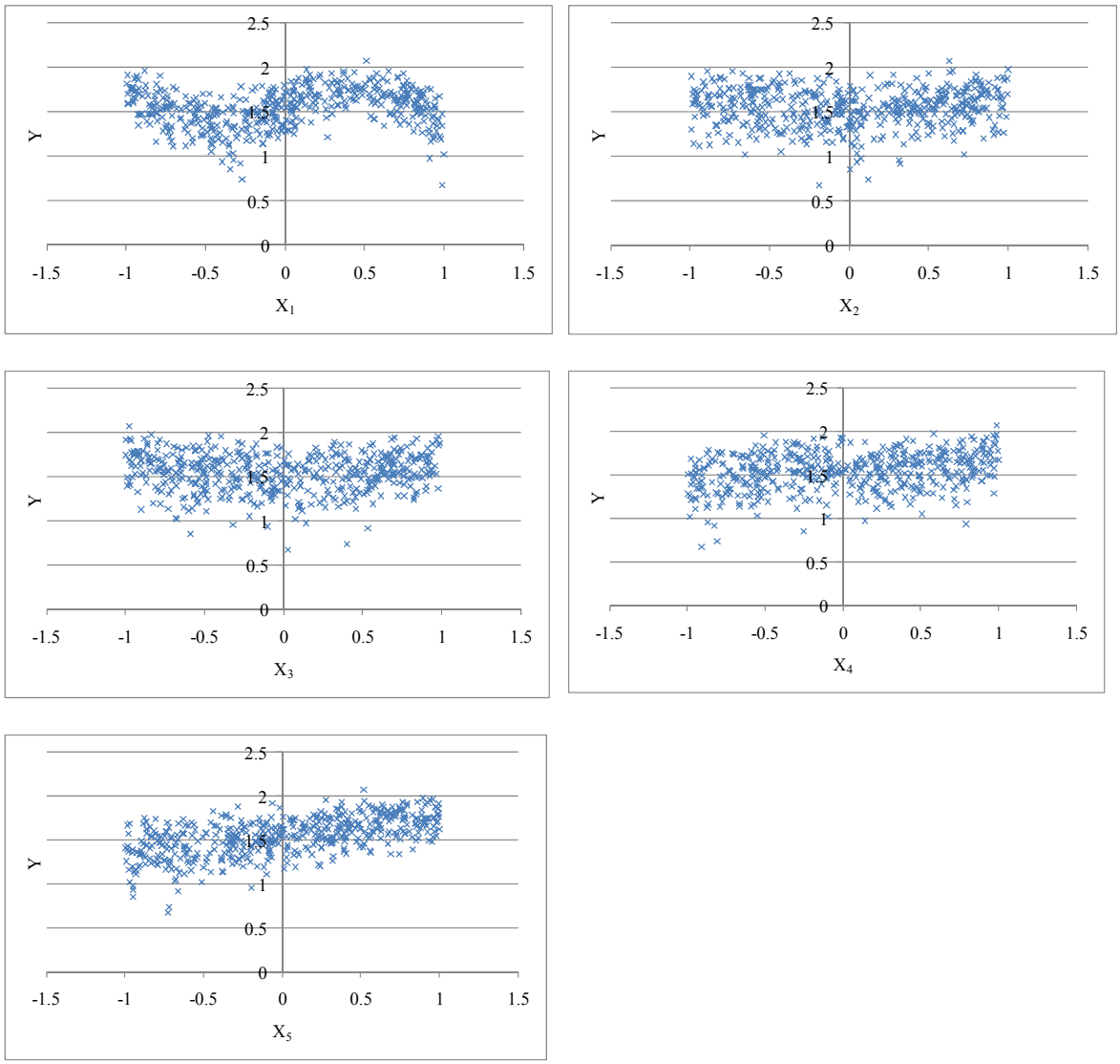

Figure 4.1. Plots of the dependent variable *Y* against each of the five independent variables X_i.

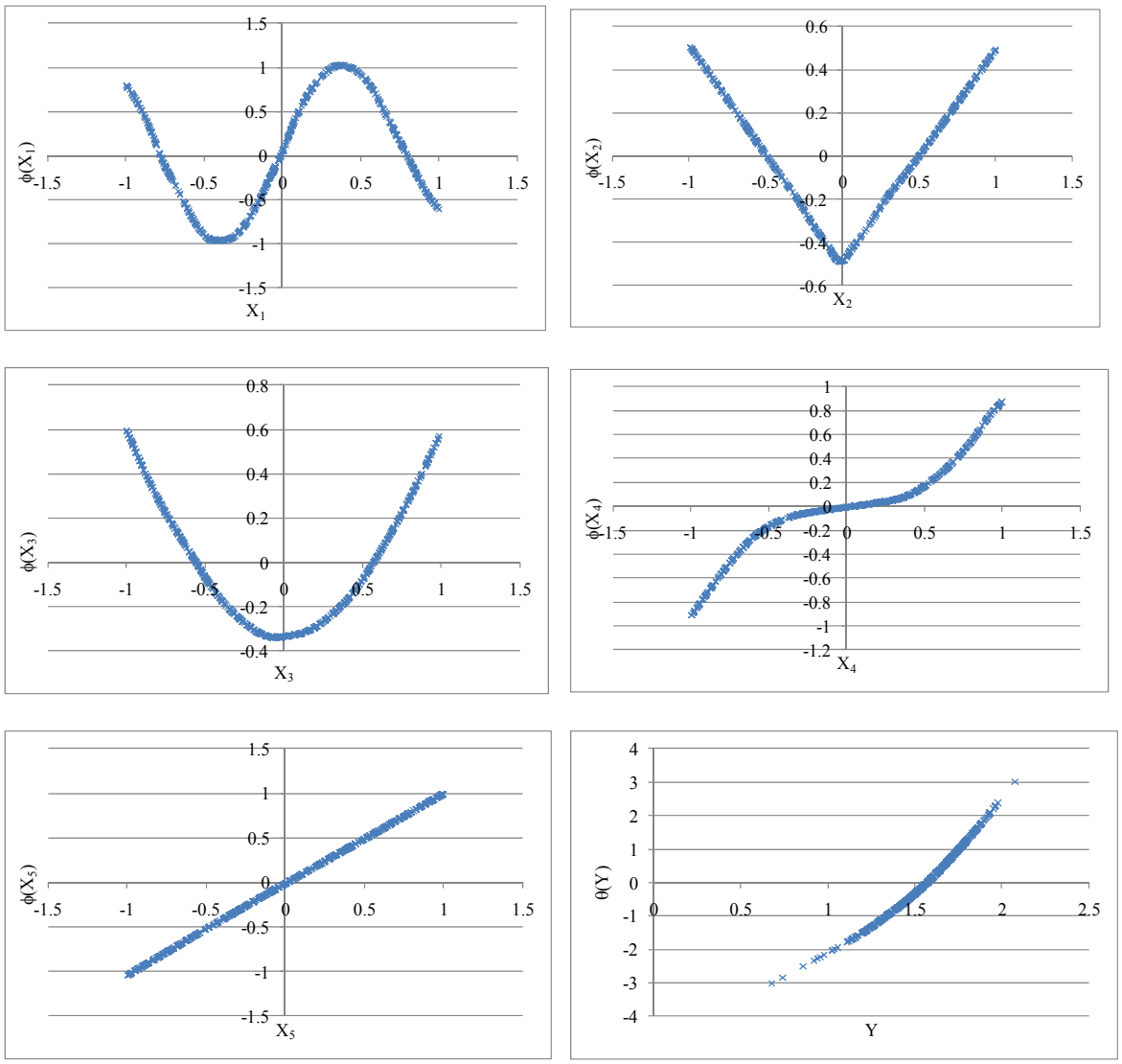

Figure 4.2. ACE tranformations of the dependent and independent variables.

From the discussion in the previous section, we know that the degree in which ACE can determine the proper correlation can be seen by examining the plot of $\theta(Y)$ vs $\sum_i \phi_i(X_i)$.

In the ideal case, the plot should be linear since that is the assumption made in the development of the theory. Figure 4.3 shows that for our example, the relationship is indeed very close to linear.

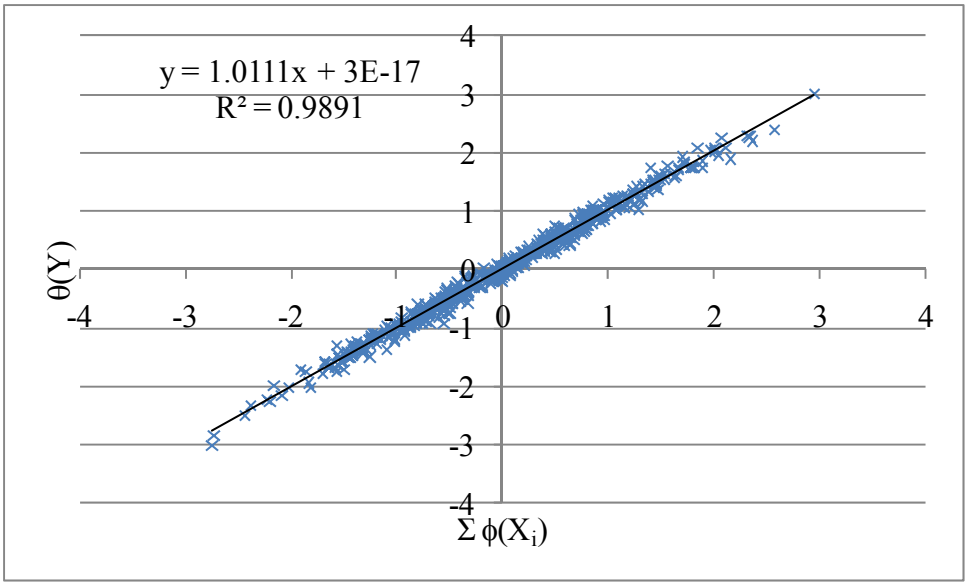

Figure 4.3. Transformed dependent variable θ as a function of the sum of the transformed independent variables ϕ_i.

CHAPTER 5

MODELING THE LOSS OF FEEDWATER TRANSIENT

5.1 Overview of the Feed and Bleed Operation

In a pressurized water reactor, the primary means of removing heat from the core during both the normal operation and after shutdown is via the steam generator (SG). Depending on the specific plant, the secondary system is composed of either 3 or 4 closed, two-phase loops that is operated at a pressure of about 800 psi (compared to ~2200 psi for the primary loop). During operation, heat generated from the core is transported via the primary water to the steam generator. The energy is transferred to the feedwater in the secondary loop causing the water to boil. The water-iiiiiiiiiiisteam mixture that is generated is then passed through steam separators which extract the liquid water back into the steam generator. The dried steam is fed to the turbine which extracts energy from the steam (this energy is converted to electricity); the steam passes through the main condenser which converts the steam into liquid water. The water is then pumped back into the steam generator via the feedwater system. The diagram of the secondary system is shown in Figure 5.1.

For the reactor during a post-trip, the flow of the steam exiting the steam generator is redirected via the main turbine bypass system and flows directly into the condenser. In the event of a loss of main feedwater in accidents such as feedwater pump failure or leak in the main feedwater loop, the steam generator will receive water from the

auxiliary feedwater (AFW) system. The AFW system draws water from various available water sources or from the condensate storage tank and feeds the water into the steam generator. However, unlike the case of the main feedwater, steam produced from the boiling of the AFW is usually vented directly into the atmosphere through the atmospheric dump valve.

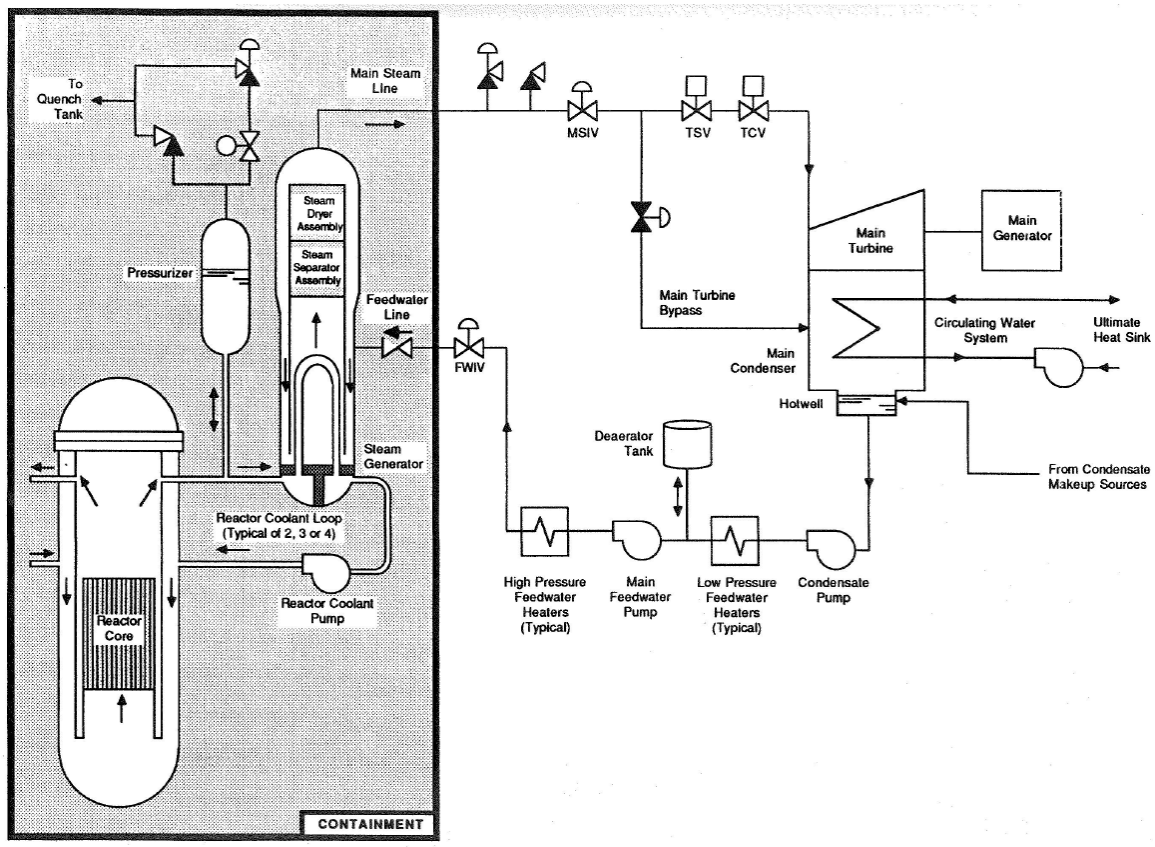

Figure 5.1. Normal pathway of feedwater flow. Diagram adapted from [NRC90a].

On the primary side, there are also systems in place to provide backup water needed for heat removal from the core. In the event of a loss of coolant (through a pipe break, for example), the emergency core cooling system (ECCS) can inject borated water from the refueling water storage tank (RWST). The ECCS system comprises three

distinct modes of water injection. If the primary system pressure is near the operating pressure, the high pressure safety injection (HPSI) pumps can supply water to the cold leg of the primary loop. These pumps can deliver water at a relatively low rate compared to the other ECCS injection modes, but can operate at a relatively high system pressure. At lower pressure, the low pressure safety injection (LPSI) pumps can provide further injection via the cold leg. There is also a third ECCS component in place called the accumulator or the safety injection tank. This is basically a pressurized tank filled with water that is connected to the primary system. When the primary pressure drops below a setpoint pressure, water will be discharged from the tank into the primary loops. Note that in our discussion in the remainder of the dissertation, the HPSI pump is also referred to as the centrifugal charging pump (CCP) while the LPSI pump is called the SI pump.

In the event of the loss of main feedwater coupled with the failure of the AFW system to actuate, the reactor will automatically scram from either the low SG level signal or high primary pressure signal. In either case, the water level in the steam generator will gradually be depleted due to boiling since the decay heat generated by the core will remain relatively high. If no action is taken by the operator, all the secondary water in the SG will eventually be depleted which will lead to a severely degraded heat removal capacity of the secondary system. The primary water will eventually reach saturation temperature and boiling will occur. Ultimately, the core will be uncovered, leading to fuel cladding damage.

To prevent the scenario described above from occurring, the feed and bleed (F&B) operation can be initiated to provide an alternate means of heat removal. This involves injecting water into the primary system via the ECCS pumps while venting the steam

from the power-operated relief valves (PORVs) located at the top of the pressurizer. The boiling that occurs in the primary will remove the decay heat. The water discharged through the PORVs is stored in the pressurizer quench tank. However, with prolonged F&B operation, the tank will become full, at which point the rupture disks on the tank will burst, releasing the water into the containment sump. In some plants, this water may be pumped back through the LPSI if necessary [Poc02]. Figure 5.2 shows the relevant pathways involved in the F&B operation.

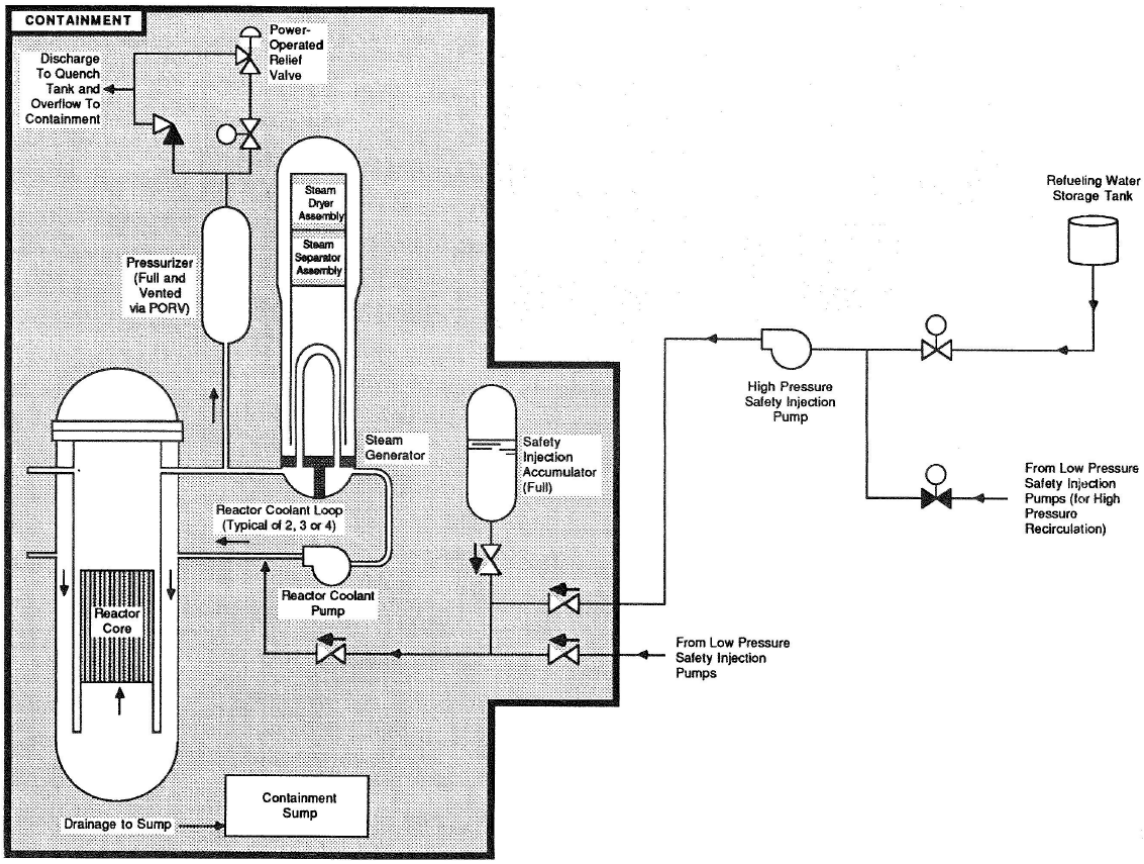

Figure 5.2. Coolant flow path during the feed and bleed operation. Diagram adapted from [NRC90a].

The F&B operation is usually done at the primary pressure near the PORV set point [Rev07]. This pressure is above the shutoff head for the LPSI pumps, therefore only

the HPSI pumps will be available. On some PWRs, even the HPSI pump will not have enough power to inject water at high pressure. In this case, the PORVs may be manually opened to relieve the pressure before the feed operation is initiated. This is known as bleed and feed [Rot94].

The emergency guideline for plant operators in the loss of heat sink event is shown in Figure 5.3. The major indicators to initiate the F&B operation include a low steam generator level (<27% of wide range level in any 3 SGs) and a high pressurizer level (pressure > 2335 psi which is the PORV set point). Note that the HPSI pump is first actuated, and if that fails, the procedure calls for the actuation of the LPSI pump. Once at least one pump is verified to be functioning, the PORVs are manually opened to provide the bleed path.

The success of the F&B operation depends on the time in which the operation is initiated and on the primary water temperature at the time of initiation [Rev07]. In addition, the number of pumps that are available for the injection, the number of PORVs that can be opened, and the injection rate will be important in determining the heat removal capacity of the F&B operation. In a static event tree that is used to analyze the success criteria of the F&B operation, the dependence of the outcome on the number of PORVs or pumps that are available for the F&B operation can be represented. However, the required feed rate depends on the primary water temperature which is a function of both the F&B initiation delay, the scram delay, and the system pressure. To include these parameters into the analysis, the ET would need to find a limit for these parameters above which the outcome is assumed to be a failure. For example, suppose that a transient simulation indicates that the F&B initiation delay time above 40 minutes will result in the

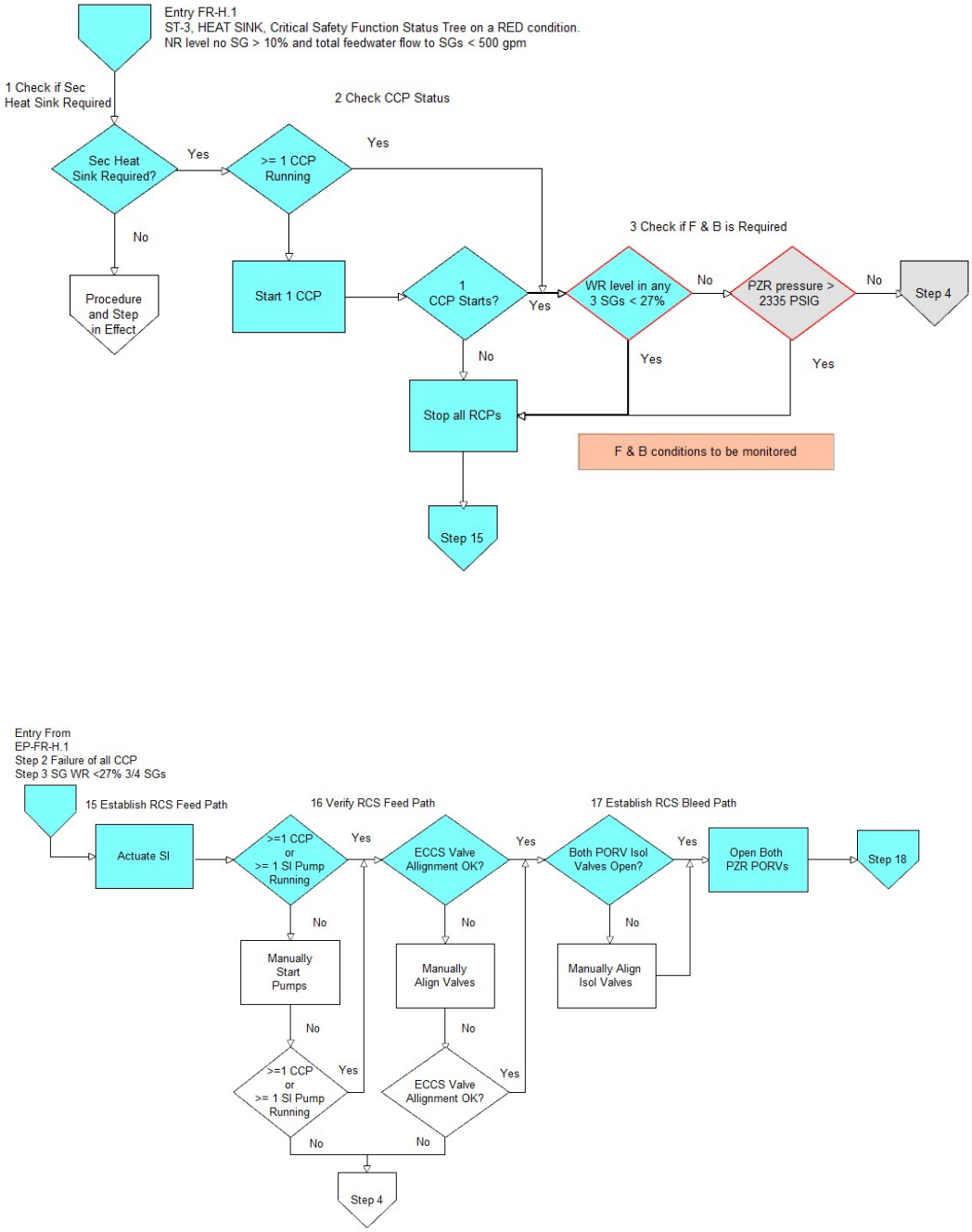

Figure 5.3. Emergency guideline associated with the loss of feedwater event. The flowchart is adapted from [Wes93].

clad temperature exceeding some safe limit. In the event tree, this criteria will be used to test the branching of the accident progression. However, if the plant condition is different from that assumed during the ET construction (for instance, suppose the reactor was not operating at full power), then the 40 minutes limit will no longer be valid. In this case, a new ET with a different branching criterion will be required.

If the ET model is aware of the plant state and changes accordingly, then the problem will have been solved. However, it is difficult for the analyst to estimate all the possible conditions of the plant and construct a risk model for every case. One way to alleviate the problem is to couple the risk model with the plant dynamic model. This way, the success or failure criteria can be updated automatically inside the model. In chapter 7, we will show how the code surrogate using ACE (which serves as the plant dynamic model) can be used in conjunction with the Bayesian network (which serves as the risk model) to perform risk assessment in a way that accounts for different plant behaviors.

5.2 The Zion Plant Description

The Zion Station-Unit 1 plant is a 4-loop pressurized water reactor of the Westinghouse design. It is located in northern Illinois and Commonwealth Edison began its operation in 1974. However, it was shut down in 1998. The plant had a rated capacity of 1095 MWe or 3411 MWth. Some basic plant data, including information on the ECCS system, is shown in Table 5.1.

The plant is equipped with three trains of AFW pumps, each with enough capacity to feed all four steam generators. Two of these trains are fed by motor driven pumps

while the third train is fed by a steam turbine driven pump. Failure of the AFW system means that none of the three trains delivers the auxiliary feedwater at their rated capacity.

One feature of the Zion-1 ECCS is that during the recirculation phase, the ECCS pump can realign its suction valve so that water is drawn from the containment sump instead of the RWST. This water is pumped into the inlet of the LPSI pumps which can be injected back into the primary system. For a prolonged F&B operation, the pressurizer sump tank will overflow and the water will be collected in the containment sump. The ability to draw water from the sump means that the available supply of the water for the injection portion of the F&B operation will not be depleted even when the RWST is empty.

Table 5.1. Some data for the Zion-1 reactor. Data from [NRC90a] and [Com82].

Thermal Capacity	3411 MWth
Electric Capacity	1095 MWe
Number of Fuel Assemblies	193
Number of Fuel Elements in Assembly	264
Number of RCS PORV	2
Number of RCS SRV	3
Number of Charging Pump	2 Centrifugal, 1 Positive Displacement
Charging System Capacity - Centrifugal	150 gpm at 2800 psig
Charging System Capacity - PD	98 gpm
Number of SI Pump	2
SI Pump Capacity	400 gpm at 1084 psig
PORV Capacity	64.6 lb/hr/MWth
PORV Lowest Setpoint	2335 psig
SRV Capacity	129,200 lb/hr
SRV Lowest Setpoint	2485 psig

CHAPTER 6

Modeling of Zion-1 Using RELAP5

This chapter will introduce RELAP5, the system analysis code that we use to simulate the loss-of-feedwater accident in the Zion-1 nuclear power station. In section 6.1, we will briefly discuss the main features, the representation of key physics, and the modeling capabilities of RELAP5 and how some these capabilities are used in our analysis. In section 6.2, we will describe the modeling of the Zion-1 plant starting from the "typical PWR" input deck that is included in the RELAP5 installation CD and the modifications that were necessary. The "typical PWR" input file itself is based on a detailed Zion-1 model, but it has some important parameters and tables removed due to proprietary considerations. This original model of Zion-1 is designed for the analysis of a loss-of-coolant (LOCA) accident, so some details that are important for the loss-of-feedwater (LOFW) transient (for instance, the pressurizer safety valves) are not included. Some additions and changes described in section 6.3 are necessary to make the deck applicable to the LOFW transient. In section 6.4, we describe the steady state and transient benchmark of our model against the published results.

6.1 Overview of RELAP5

RELAP5/MOD3.3 (Reactor Excursion and Loss of Coolant Analysis Program) [NRC01] is a one-dimensional analysis code designed for the analysis of the transient behaviors of

light water reactors during postulated accident conditions. The code allows the modeler to construct hydrodynamic volumes which can be connected via different types of junction with a prescribed boundary condition such as flow rate or temperature. Conductive heat transfer between several adjacent (but hydrodynamically isolated) volumes can be specified via the use of heat structure. Heat structures are also used in the solid structure that generates heat such as in the fuel rod or pressurizer heater. Using the appropriate connections of the hydrodynamics volume and heat structure, both the primary and secondary systems of the light water reactor can be modeled.

Controls of the reactor component can be achieved by the use of trip cards. Trips are basically logic statements that specify when the trip will be true based on a set of inputs. The input to a trip statement can be either other trips or variables that describe the state of a plant component. For instance, a trip logic can be used to represent the pressure where the valve should open or the point in time when a pump should start to coast down. Through the use of appropriate trip combinations, a sophisticated set of control functions and settings can be specified in the RELAP5 model.

Internally, the RELAP5 modeling is broken into the fluid model, the heat transfer model, and the reactor kinetics mode. The code uses a two-fluid, six-equation model to describe the fluid behaviors. The two-fluid model allows the treatment of the water/steam mixture and any non-condensable gases that may be present. For reactor kinetics, a point kinetics model is used. Point kinetics treats the neutron flux as a product of the space-dependent and the time-dependent functions. The flux shape function is assumed to be constant in time, while the function describing the magnitude of the flux is time dependent. Heat conduction is modeled using the standard integral formulation of the

conduction equation. In addition to the three physics models mentioned above, RELAP5 also allows analysts to build control systems which will automatically control certain plant components such as valves and heaters based on plant variables. Unlike trips whose output are binary, a controller allows the specification of the boundary condition at a node based on values at another node. For instance, the flow rate of the feedwater can be automatically adjusted based on the mass balance of the steam generator. This capability for modeling of the control system allows automatic plant controls and equipment setpoints to be accurately represented.

6.2 Modeling of Relevant Physics in RELAP5

This section will briefly mention some of the main physics models that are used in RELAP5. Since most of these models use the volume and time-averaged value of the variables as the parameter, all the variables listed in this section are cell-averaged quantities and will not be explicitly designated in the individual equations.

6.2.1 Conservation Equations

The description and equations presented in this section are adapted from [Ran89] and [NRC01]. RELAP5 uses a six-equation model to represent the behavior of the fluid. These six equations represent the conservation of mass, momentum, and energy for each of the two phases of the fluid. The conservation of mass for the k^{th} phase is represented by:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k) + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_k \rho_k v_k A) = \Gamma_k, \tag{6.1}$$

where

$\alpha_k =$ volume fraction of phase k,
$\rho_k =$ density of phase k,
$v_k =$ fluid velocity of phase k,
$A =$ cross sectional area, and
$\Gamma_k =$ mass generation rate per unit volume for phase k.

Here, the subscript $k \in \{f, g\}$ denotes either the fluid or the gas phase. The first term on the LHS represents the change in mass due to a change in density. This change may be due to the thermal expansion of the coolant following an increase in temperature or a contraction due to a pressure drop after a relief valve is opened. The second term describes the balance between mass inflow and outflow from a unit volume. The mass generation term accounts for phase change (boiling or condensation) of the coolant. Note that since there can only be two phases in Eq. (6.1), we also require:

$$\Gamma_f = -\Gamma_g. \tag{6.2}$$

Generally, the phasic momentum equations are a set of three equations, representing each of the three directions in a 3-dimensional space. However, with the assumption that the flow is only aligned in one direction, we can neglect the average velocity in the other two coordinates. The conservation of momentum equation in the flow direction becomes, for the gas phase:

$$\begin{aligned} \alpha_g \rho_g A \frac{\partial v_g}{dt} + \frac{1}{2}\alpha_g \rho_g A \frac{\partial v_g^2}{\partial x} = & -\alpha_g A \frac{\partial P}{\partial x} + \alpha_g \rho_g B_X A - \left(\alpha_g \rho_g A\right) \mathrm{FWG}(v_g) \\ & + \Gamma_g A (v_{gI} - v_g) - \left(\alpha_g \rho_g A\right) \mathrm{FIG}(v_g - v_f) \\ & - C \alpha_g \alpha_f \rho A \frac{\partial \left(v_g - v_f\right)}{\partial t} + v_f \frac{\partial v_g}{\partial x} - v_g \frac{\partial v_f}{\partial x}, \end{aligned} \tag{6.3}$$

and for the liquid phase:

$$\alpha_f \rho_f A \frac{\partial v_f}{dt} + \frac{1}{2}\alpha_f \rho_f A \frac{\partial v_f^2}{\partial x} = -\alpha_f A \frac{\partial P}{\partial x} + \alpha_f \rho_f B_x A - \left(\alpha_f \rho_f A\right) \mathrm{FWF}(v_f)$$
$$+\Gamma_f A(v_{fI} - v_f) - \left(\alpha_f \rho_f A\right) \mathrm{FIF}(v_f - v_g) \qquad (6.4)$$
$$-C\alpha_f \alpha_g \rho A \frac{\partial\left(v_f - v_g\right)}{\partial t} + v_g \frac{\partial v_f}{\partial x} - v_f \frac{\partial v_g}{\partial x},$$

where

P = pressure,
C = coefficient of virtual mass,
FWG, FWF = wall frictional drag coefficient for the gas and liquid phases, respectively,
FIG, FIF = interphase drag coefficient for the gas and liquid phases, respectively,
B_x = body force in the flow direction
v_{gI}, v_{fI} = interfacial velocity for the gas and liquid phases, respectively.

The terms on the right side of Eqs. (6.3) and (6.4) represent, in the order shown in the equations, the pressure gradient, the body force (such as from pump head), wall friction, momentum transfer due to interphase mass transfer, interphase friction drag, and the force due to mass acceleration (also called the virtual mass). Note that the interphase drag coefficients are linear in the relative phase velocity.

The thermal energy conservation equation is:

$$\frac{\partial}{\partial t}(\alpha_k \rho_k U_k) + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_k \rho_k U_k v_k A) = -P\frac{\partial \alpha_k}{\partial t} - \frac{P}{A}\frac{\partial}{\partial x}(\alpha_k v_k A) + Q_{wk} + Q_{ik}$$
$$+\Gamma_{ig} h_k^* + \Gamma_w h_k^{'} + \mathrm{DISS}_k, \qquad (6.5)$$

where

Q_{wf}, Q_{wg} = phasic wall heat transfer rates per unit volume
h_g^*, h_f^* = phasic enthalpies associated with bulk interface mass transfer
$h_g^{'}, h_f^{'}$ = phasic enthalpies associated with wall interface mass transfer
Γ_{ig} = vapor generation from energy exchange
Γ_w = vapor generation due to wall heat transfer effects
Q_{if}, Q_{ig} = interface heat transfer
$DISS_f, DISS_g$ = phasic energy dissipation (wall friction, pump effects).

The first term on the LHS describes the heat added into the control volume (for instance, through a heat structure). The second term represents the net heat flow into the control volume by mean of the motion of the fluid (convective heat transfer). The RHS describes the various looses due to friction or the transfer of energy between the two phases.

6.2.2 Heat Conduction Representation

A heat conduction model is used in RELAP5 to calculate the heat transfer across the solid portion of the reactor components such as across the fuel cladding and across the U-tubes carrying the primary coolant inside the steam generator. The integral form of the heat conduction equation is:

$$\int \rho(x,T)\frac{\partial T(x,t)}{\partial t}dV = \int k(x,T)\nabla T(x,t)\bullet dA + \int S(x,t)dV, \qquad (6.6)$$

where

ρ = volumetric heat capacity,
k = thermal conductivity,
T = temperature,
S = heat source.

The surface heat flux $q"$ required to specify the boundary condition can be calculated as:

$$q'' = h(T - T_{surface}) = -k\frac{\partial T}{\partial z}, \tag{6.7}$$

where it is assumed that the *z* vector is the outward normal vector relative to the surface of the boundary. RELAP5 has several correlations built into the code which are used to calculate the heat transfer coefficient *h*, depending on the flow condition of the hydrodynamics volume. Once the appropriate flow type is selected by the code and the heat transfer coefficient calculated, the surface heat flux in Eq. (6.7) is used as a boundary condition to the heat conduction equation (6.6).

6.3 Modeling of the Zion Plant

Figure 6.1 shows the nodalization of the hydrodynamic volumes of the "typical PWR" input deck that is included as a sample input deck in the RELAP5 installation. The reactor pressure vessel is modeled with 15 volumes comprising the inlet annulus (volumes 300, 305, 310), the downcomer, the downcomer bypass (volumes 315, 320), the lower plenum (volumes 322-325, 330), the upper plenum (volumes 340, 345, 350), the upper head (volumes 355, 356), and the core (volume 335).

The core volume is represented with a PIPE component and is subdivided into 6 nodes. Two heat structures are connected to volume 335. The first heat structure, divided into 6 axial and 2 radial segments, models the core barrel. This structure connects volume 335 with the downcomer bypass represented by volume 320. No heat generation is present in this heat structure. The second heat structure represents the fuel pins. It is divided into 6 axial and 16 radial segments. The 16 radial segments represent the fuel material (6 segments), the air gap (2 segments), and the fuel cladding (8 segments).

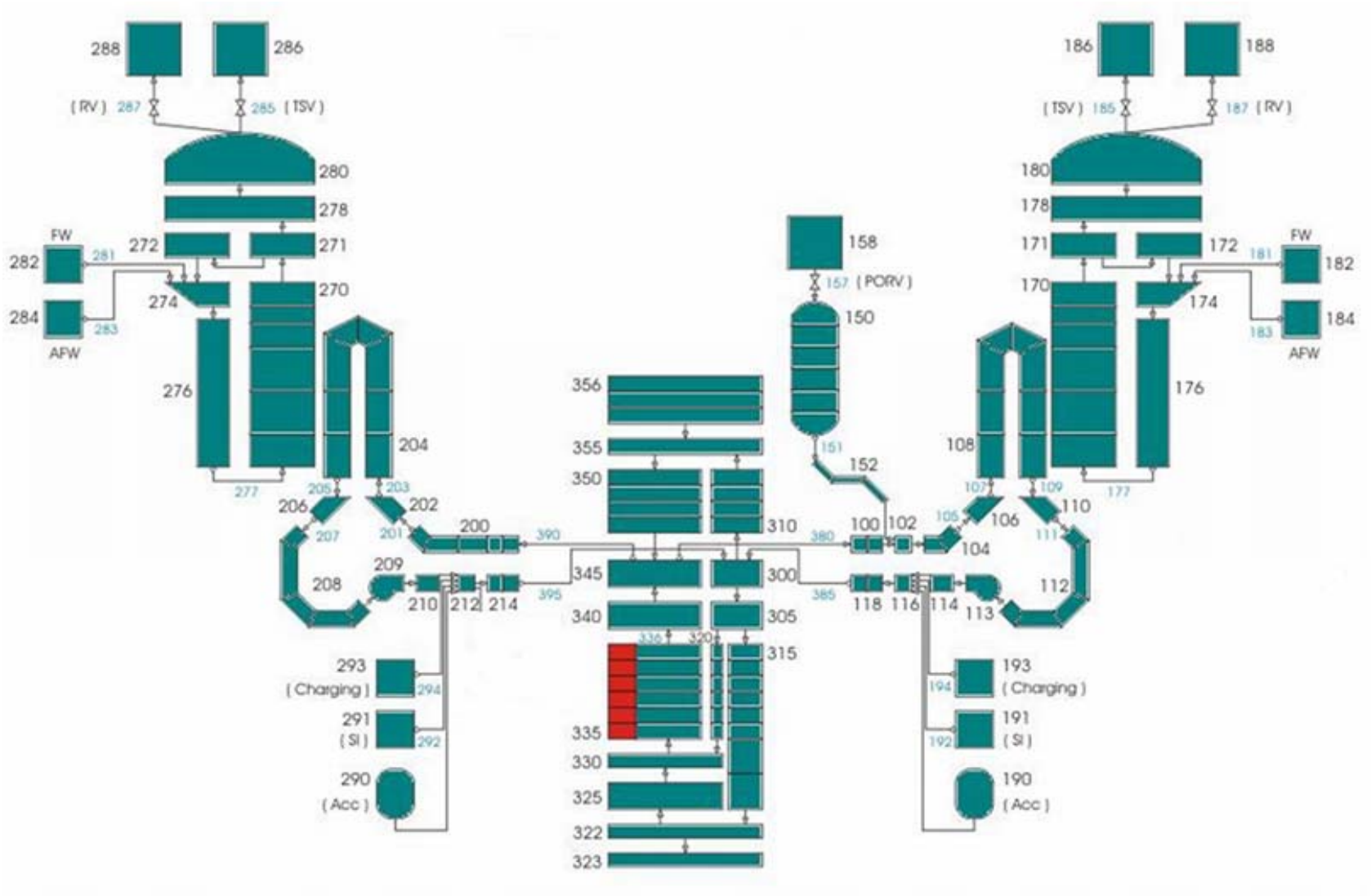

Figure 6.1. Nodalization diagram for a representative pressurized water reactor.

Even though Zion-1 is a 4-loop reactor, the "typical PWR" model only has two loops. One of the loops, labeled as the "intact loop", actually represents a merging of the 3 physical loops. The "broken loop" is one physical loop. This means that parameters such as flow area and flow rate for the "intact loop" is three times that of the "broken loop". (The reason for this merging is that the original deck is designed for a LOCA analysis and so the loops that do not contain the break are merged to reduce run time.) Except for the pressurizer, the layout of the broken and intact loop is identical.

The pressurizer, represented by volume 150, is a PIPE component that is divided into 6 segments. It is connected to the hot leg of the intact loop (it is moved to the broken loop in the final model) via the surge line (volume 152). The top portion of the pressurizer contains an outlet to the PORV (represented by a trip valve 157) which drains into volume 158, representing the quench tank.

For each of the steam generators in the two loops, the U-tubes (volume 204 and 108) are modeled with a PIPE component and are each divided into 8 segments. The loss coefficients and hydraulic diameters for these volumes are explicitly specified using appropriate values so that the total flow rate and area computed by the code agree with plant data.

The ECCS is represented by two charging pumps (volumes 193 and 293), two LPSI pumps (volumes 191 and 291), and two accumulators (volumes 190 and 290). The sources of the water for all the ECCS pumps are assumed to be infinite for the purpose of the simulation. The dependence of the flow capacity of both the HPSI and LPSI systems are specified via the time-dependent junction which specifies the inject rate as a function

of the primary pressure. These junctions are activated by the appropriate trip signal that can be triggered either manually or automatically via a low pressure signal.

The secondary side is modeled by 12 volumes. Volumes 170 and 270 have a heat structure connection to volumes 108 and 204, respectively. This connection represents the heat exchanger portion of the steam generator. The heat structure connecting the primary and secondary systems has 16 axial and 8 radial segments. The main feedwater (volumes 182 and 282) and the AFW (volumes 184 and 284) are connected to the steam generator via a branch node (volumes 174 and 274). As in the case of the ECCS, the source of water for these systems is assumed to be infinite. The steam dome is represented by two volumes for each steam generator: a BRANCH component (volumes 180 and 280), and a PIPE volume (volumes 178 and 278) .The steam separator is modeled by the steam separator component (volumes 171 and 271).

Five materials are defined in the model: the uranium oxide fuel, the air gap, zircaloy-4 (fuel clad), inconel (U-tubes), and stainless steel (structural components). Each of these materials has a corresponding table for the temperature-dependent thermal conductivity and specific heat capacity.

We made modifications to the typical PWR input deck described in the preceding paragraphs to improve the agreement between our model and the data in various published reports for both the steady-state benchmarking and the LOFW transient scenario.

1. The first modification is the addition of the pressurizer heater and spray line. In the original input file, these two components necessary for pressure control are absent and we noticed that it was difficult to achieve a steady state with the correct pressure. The spray

line is added so that it draws water from the cold leg (volume 212) and sprays water from the top of the pressurizer (segment 1 of volume 150). Both the heater and the spray are controlled via a proportional controller which takes input from the pressure error at the pressurizer volume.

2. The second modification involves moving the pressurizer from the "intact loop" to the "broken loop". Recall that the intact loop is really three physical loops merged into one. We felt that the behavior of the pressurizer will be more accurate if it is connected to a loop whose flow condition is closer to the physical loop.

3. The original deck has one PORV to represent the two PORVs that are actually present in the plant. However, for the LOFW transient, there are situations where the PORVs do not have sufficient relief capacity to drain the expanding primary coolant which leads to an increase in the primary pressure large enough so that the code safety valves open. These safety valves were not originally included in the model and had to be added. The control setpoints for the PORVs and the safety relief valves are shown in Table 6.1.

Table 6.1. Setpoints for primary relief valve operation.

	Setpoint Pressure (psia)
PORV open	2350
PORV close	2330
Relief valve open	2535
Relief valve close	2460

4. The control systems for the feedwater system, the steam dump valves, the emergency injection system, and the PORVs were also modified to make them more consistent with the actual behavior of these components. Trip controls were attached to these components so that they actuate when control setpoints are reached. We also had to decrease the pump

velocity by 3% to get a better agreement with the steady state values in the final safety analysis report (FSAR) [Com82].

5. Finally, the control parameters for the steam separator in the steam generator were adjusted to make the separator function properly in a low flow rate situation. The VOVER parameter specifies the void fraction above which the steam will be treated as pure vapor. Similarly, the VUNDER parameter specifies the void fraction limit below which the steam-water mixture will be treated as pure liquid. During the LOFW transient, the steam flow rate drops low enough so that the default values of these parameters were not adequate. We decreased the limit to 1.0E-3 for both parameters.

6.4 Benchmarking of the RELAP5 Model

The benchmarking for both the steady-state case and the LOFW transient case is done to ensure that our Zion-1 model adequately represents the transient behavior during the accident. In the tables and description below, our Zion-1 model will be referred to as the TPWR model. The steady state comparison of the TPWR model and the values from the FSAR is shown in Table 6.2. We see that for most parameters, the agreement is within 1% of the FSAR values.

Figures 6.2 through 6.6 compare the LOFW transient results between the TPWR model and the values obtained using the TRAC code as presented in [Dem82]. (The FSAR does not have results for the complete LOFW transient.) Table 6.3 lists the sequence of event for the case used in this benchmarking. The transient starts with the loss of main feedwater, followed by the reactor coolant pump trip and the reactor scram at 1 second. The AFW should come on 15 seconds after the loss of main feedwater but in

this case, it fails to do so. At 60 seconds into the transient, the atmospheric relief valve on the secondary side opens in response to high steam generator pressure. Note that the closing of the main steam isolation valve occurs coincident with the reactor scram.

Table 6.2. Comparison of key parameters between the RELAP5 model and the FSAR.

	FSAR	TPWR Model
Vessel Inlet Temp (F)	530	529
Vessel Outlet Temp (F)	594	593
Coolant Flow Rate (lbm/s/loop)	9374	9431
Pressurizer Pressure (psia)	2248	2252
Pressurizer Temp (F)	653	653
SG Steam Exit Temp (F)	506	505
SG Pressure, shell side (psia)	719	719
SG Steam Exit Flow Rate (lbm/s)	972	971

Table 6.3. Sequence of event for the loss-of-feedwater transient benchmark.

Time (s)	Event
0.0	Loss of main feedwater
1	RCP trip
1	Reactor scram
15	AFW system fails
60	Atmospheric relief valves open

Figure 6.2 shows the reactor power as a function of time. Both the TPWR model and the TRAC model that is used in [Dem82] use the American Nuclear Society Standard ANS 5.1 to represent the decay power generated by fission products so we do not expect too much difference in the decay power between the two models. Figure 6.3 shows the average reactor coolant temperature as a function of time. After the RCP trip at 1 second, the primary coolant flow rate decreases as the coolant pumps gradually coast down to about 5% of the flow rate at full power (Figure 6.5). This drop in flow rate will cause the

temperature of the coolant to briefly increase (this behavior is called the power-to-flow mismatch in [Dem82]). However, by about 10 minutes into the transient, natural circulation will become the dominant form of primary coolant flow and the temperature will reach its quasi-steady state value. During this time, the primary means of heat removal is through the boiling of the water inventory that remains in the steam generator after the loss of main feedwater. By 80 minutes, the steam generator will have depleted its water inventory and the primary coolant temperature will increase until it reaches saturation temperature. In the period between the dryout of the steam generator and the boiling of the primary coolant, we see an increase of the volume of the coolant due to thermal expansion. This is shown in Figure 6.6, where the pressurizer becomes solid about 100 minutes into the transient.

For the most part, the behavior of the reactor during the transient as calculated by our model agrees with the prediction from [Dem82]. The time of the steam dryout is different by about 10-20 minutes but we feel that this difference is small enough so that we can assume our model can adequately model the behavior of Zion-1 during a loss of feedwater transient.

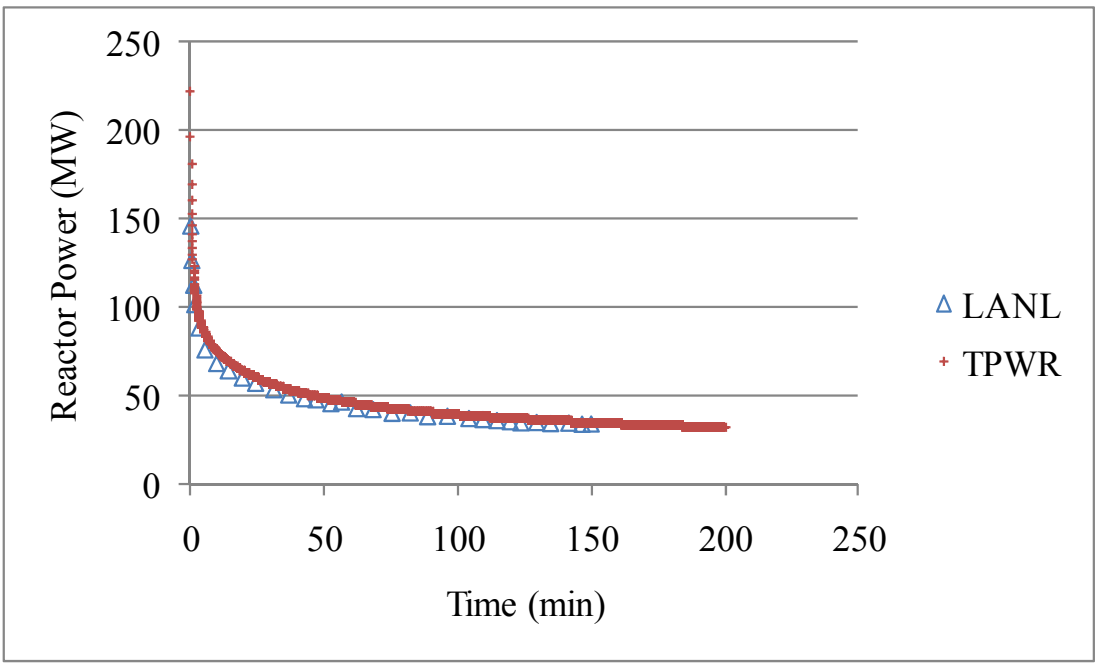

Figure 6.2. Reactor power as a function of transient time.

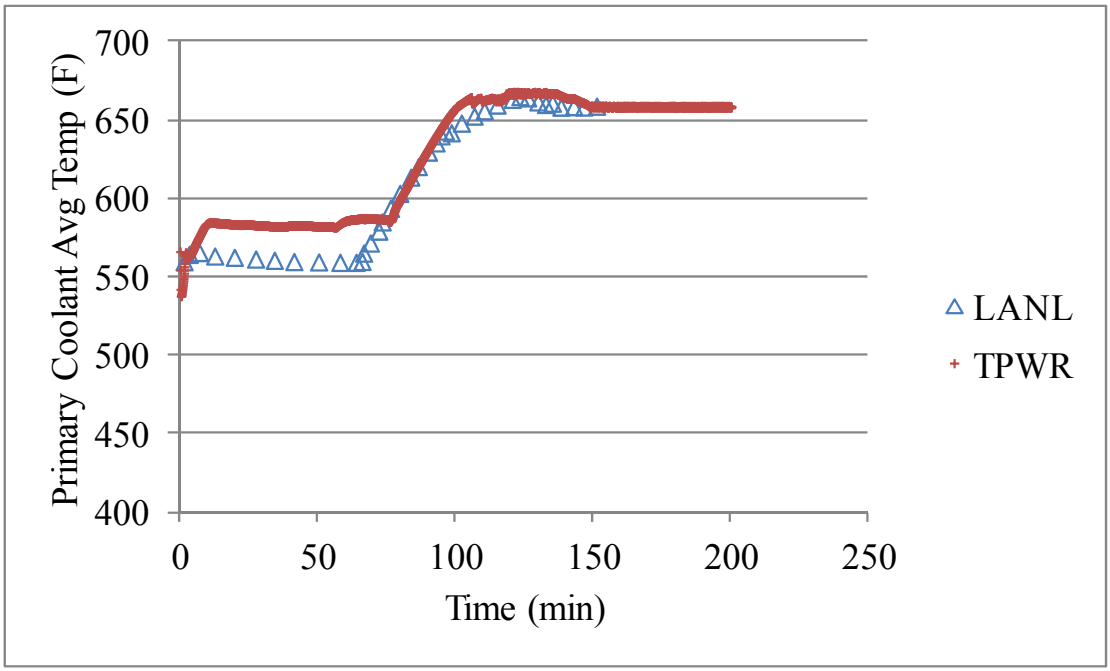

Figure 6.3. Average coolant temperature as a function of transient time.

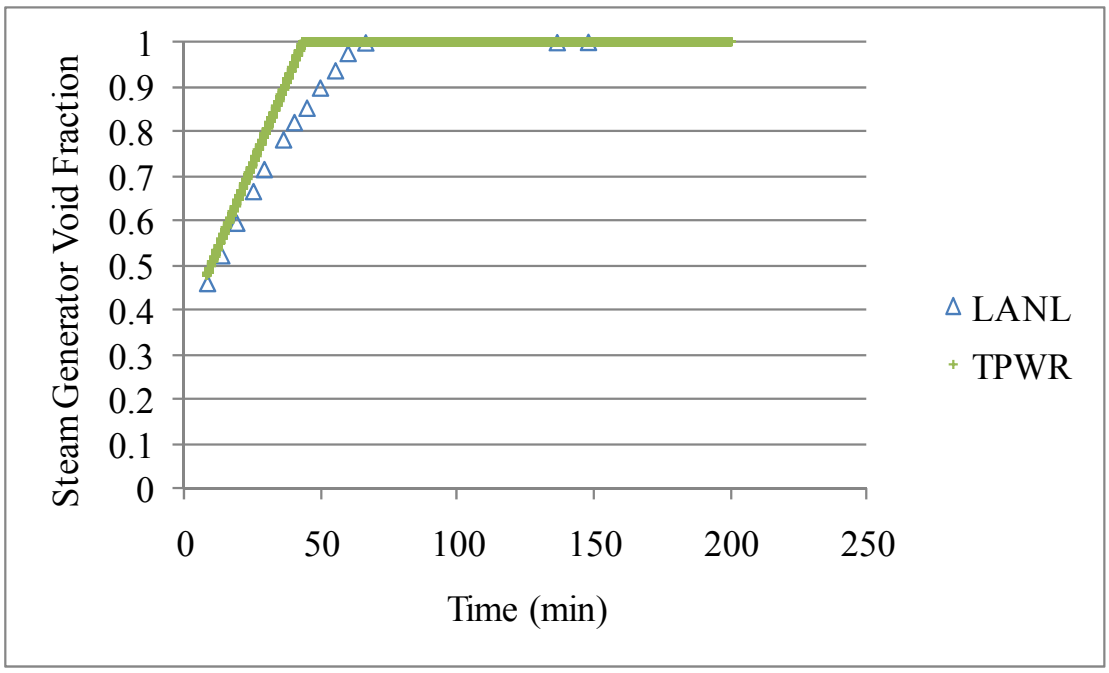

Figure 6.4. Steam generator void fraction (shell side) as a function of transient time.

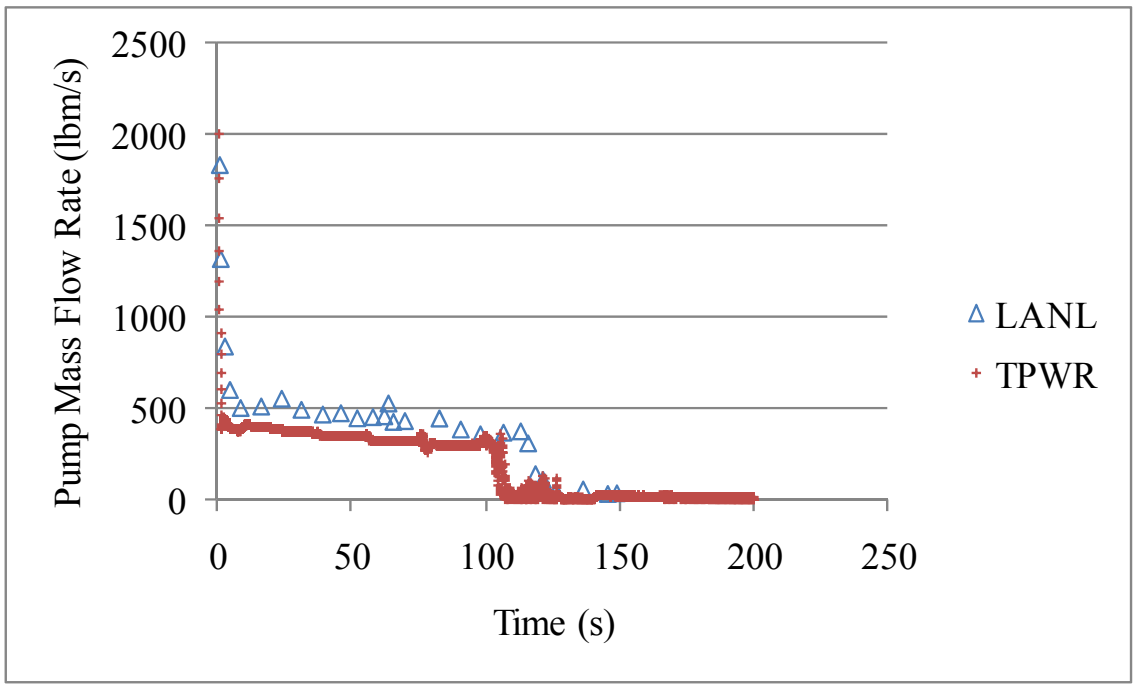

Figure 6.5. Primary coolant mass flow rate as a function of transient time.

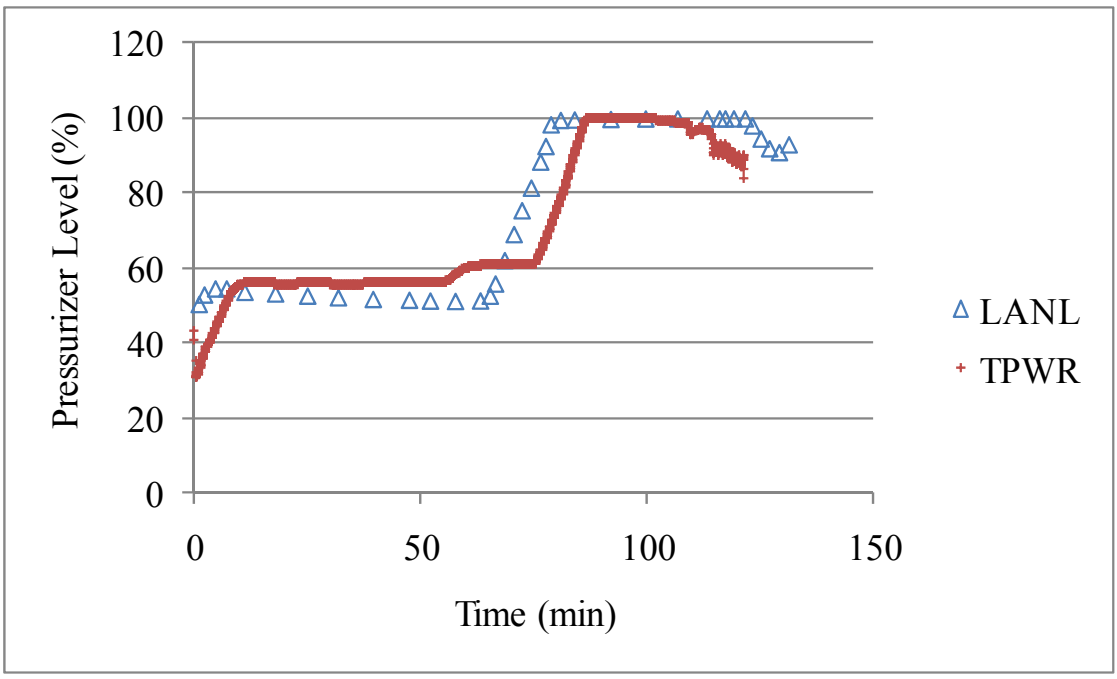

Figure 6.6. Pressurizer level as a function of transient time.

CHAPTER 7

DYNAMIC RISK ANALYSIS FOR THE LOSS-OF-FEEDWATER TRANSIENT

In this chapter, we will present the analysis results of loss-of-feedwater transients coupled with the implementation of the F&B operating procedure for the Zion-1 plant. The objective of the analysis is to calculate the time-dependent core damage probability resulting from the LOFW transient. The RELAP5 model used for the analysis is described in chapter 6. Unlike the water tank example of Chapter 2, we will consider several independent variables simultaneously. The number of independent variables presents a problem in that an adequate sampling of all the possible combinations of these variables will take a prohibitively large number of code runs. However, using the appropriate conditional independence assumption, we can construct a Bayesian network model which will allow us to reduce the number of RELAP5 cases necessary for good sampling to a more manageable level.

The determination of the conditional independence relationships within a set of variables can be a subjective exercise. As we have shown in Chapter 3, conditional independence and (pair-wise and joint) independence do not imply one another. This means that even for variables that are independent, conditional independence may not hold given some other variable. Complicating the matter further, for a given set of conditional independent variables, it is often possible to construct a Bayesian network in many different ways, depending on the selection of the root nodes. In cases like this, the

choice of the appropriate network will be reduced to selecting a model that is the most intuitive given the knowledge of the physics of the problem.

7.1 The Dynamic Bayesian Network Model for Feed and Bleed Operation

The construction a DBN model requires us to make certain assumptions about the conditional independence relations of relevant plant equipment. As we have mentioned in Chapter 3, a DBN model that is constructed without exploiting conditional independencies will have little advantage over other common time-dependent analysis techniques such as the Markov chain in terms of efficiency. Even in situations where it is difficult to make a blanket statement about the conditional independency of a set of variables, we may still be able to construct a useful DBN model if we limit ourselves to an appropriate subset of the state space where the conditional independence assumption does apply (or at least, where conditional dependency is weak enough to ignore).

The ability to define suitable sub-region in the state space where a particular DBN model is valid is exploited in the construction of the network shown in Figure 7.1. The dynamic network model in the figure is expanded over two time slices for clarity. Recall from Chapter 3 that one time slice may be thought of as one time step in the discretization of the time variable. For instance, slice 0 can be any time step n while slice 1 is the next time step $n+1$. We should again note that unlike in numerical analysis, the size of the time step does not play a role in a DBN model. Rather, time difference between any two slice is only important when constructing the inter-slice transition probability.

The DBN model in Figure 7.1 has the variable *core damage* as its only leaf node. For simplicity, we will assume that this node has only two states: *failed* (representing core damage), and *OK* (representing no core damage). We are interested in the probability that the node will be in the *failed* state at any given time during the transient. The parent of *Core Damage* is the node *Clad Temp*. The interpretation of this statement is that whether or not core damage occurs depends only on the maximum clad temperature. Note that even though this assumption may not be valid in the general sense (other factors may affect the chance of core damage in other types of accidents), high clad temperature is the dominant indicator of core damage in the LOFW transient which justifies the assumption. The next step is to specify quantitative relationships between the root and child nodes via appropriate conditional probability functions. From the Final Acceptance Criteria [NRC74], the upper temperature limit for Zircaloy clad before clad damage occurs is 2200^{o}F. This motivates our next assumption that any temperature higher than this limit will result in core damage with a probability of 1.0. We will also assume that clad temperatures below 1600^{o}F will not cause core damage. The damage probability increases linearly between these two limits. The relationship described here is shown in Figure 7.2. The triangular probability density function represents the uncertainty in the core damage probability. The corresponding cumulative distribution function is also represented for the RHS ordinate of Figure 7.2.

Moving on to the next level in Figure 7.1, we see that *Clad Temp* has three parents: the primary coolant temperature, primary coolant pressure, and the coolant flow rate. This selection of parent variables will be validated in Section 7.3 when we construct the ACE model for *Clad Temp*. Unlike the case of the *Core Damage* node, the

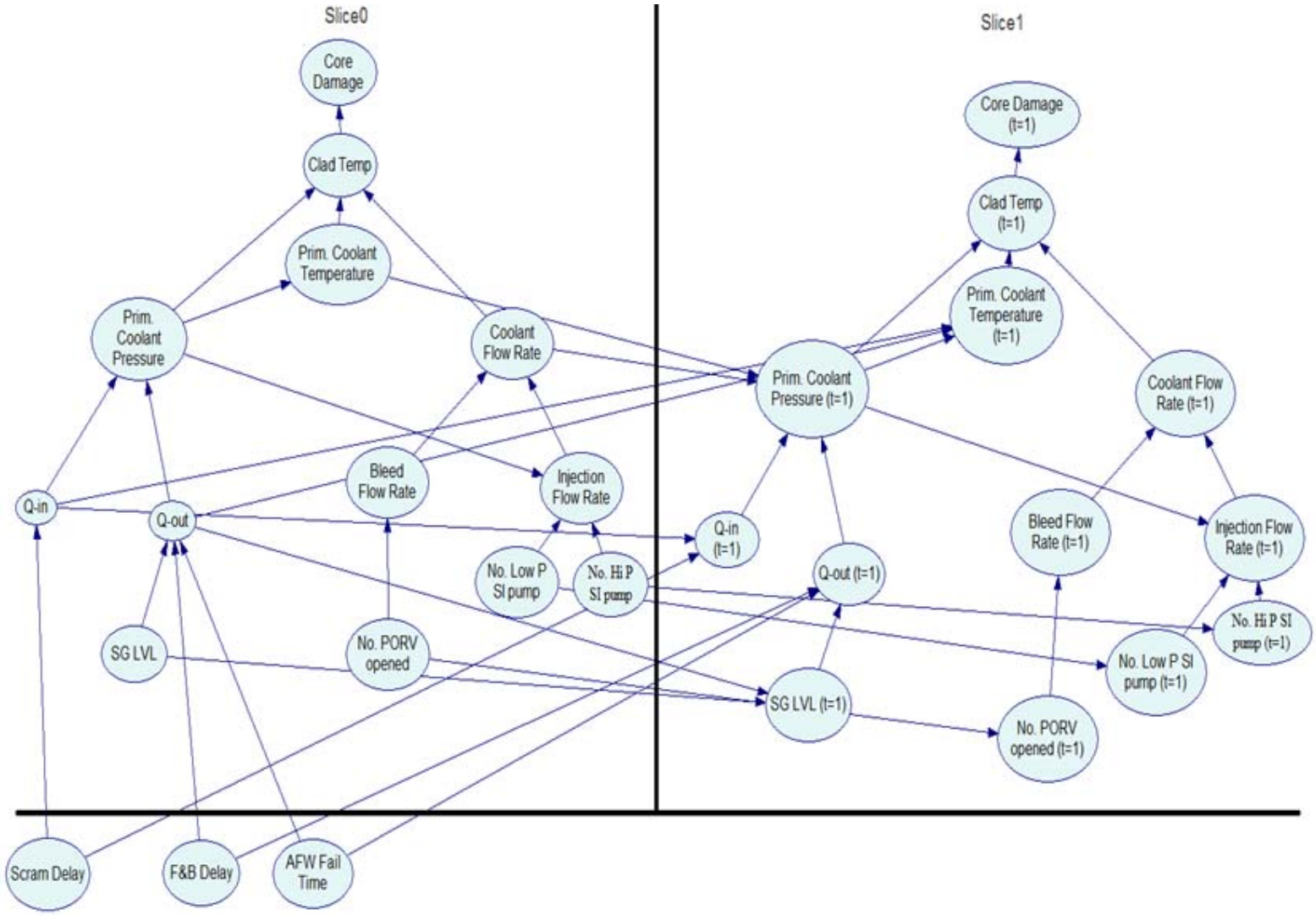

Figure 7.1. Bayesian network model for analyzing the F&B scenario. The graph is unrolled over two time slices.

conditional probability function for *Clad Temp* is deterministic in the sense that given a value for each of the three parent variables, we assume that we may determine the clad temperature uniquely. Therefore, the conditional "probability" function needs to be constructed based on this deterministic relationship. For our model, this deterministic function is obtained via the ACE regression. As an aside, we note that even though the relationship is deterministic, the use of a probability function also allows us to use "noisy" or "leak" probability which can be used if *Clad Temp* has additional parents which are not modeled.

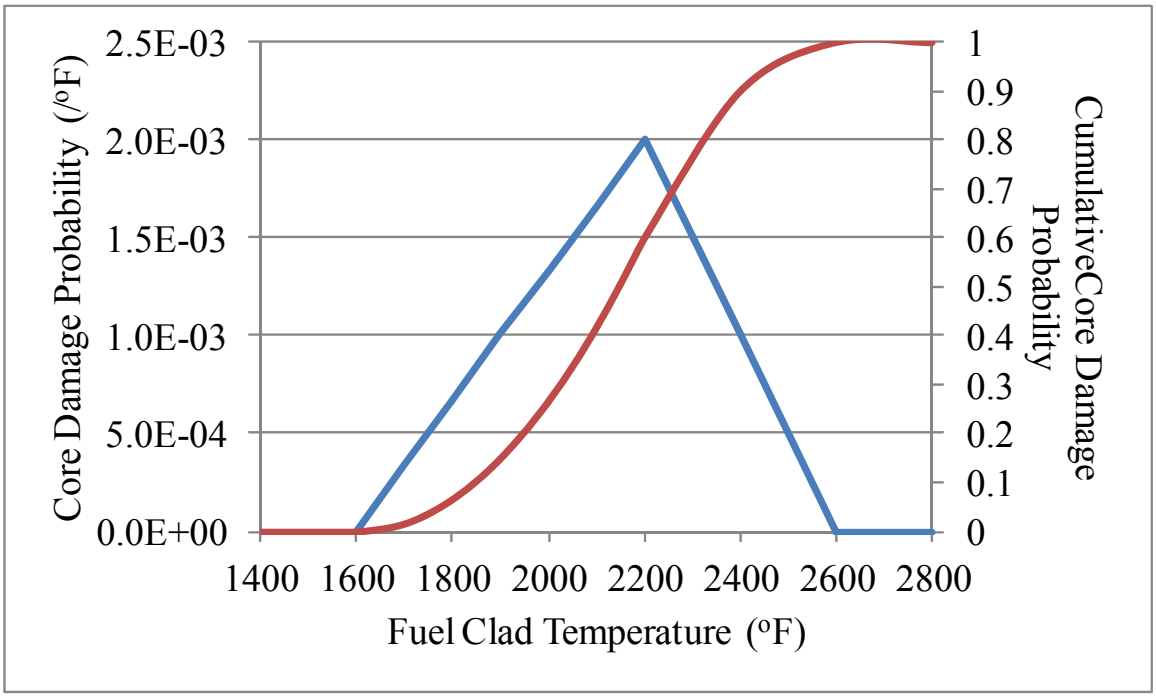

Figure 7.2. Relationship between the maximum fuel clad temperature and the probability of core damage.

The node *Injection Flow Rate* represents the rate at which the injection system (both the high pressure and low pressure injection system) pumps water into the primary system for the feed portion of the F&B operation. The injection rate depends on the

number of pumps operating (represented by the nodes *No. Low P SI pump* and *No. Hi P SI pump*) as well as the pressure of the primary system. The dependence on the primary pressure arises from the pressure-dependent pumping capacity for the two pump types. As mentioned in Chapter 6, the Zion plant has two types of pump that can be used to inject water into the primary system during normal operation. The high pressure centrifugal charging pump can inject water at a higher primary system pressure but with a relatively low flow rate. The low pressure injection pump has a lower shutoff pressure but can inject water at a higher rate. The pressure dependent flow rates of the these two pump types are shown in Table 7.1.

Table 7.1 Flow rate capacity for the low and high pressure injection pumps.

Primary pressure (psia)	Low pressure SI pump	High pressure charging pump
	Liquid Injection Rate (lbm/s)	
2620	0	2.65
1685	0.333	7.03
1529	1.00	7.43
839	9.00	9.30
15	53.6	9.70

Figure 7.1 show that the nodes *Low P SI pump* and *No. Hi P SI pump* are also parents of the corresponding node in the next time slice. This is represented in the network by arcs originating from slice 0 pointing to nodes in slice 1. The interslice dependency is introduced into the model to account for possible failure of the pumps. For the sake of discussion, let us assume that one time slice represents 1 hour. If the failure rate of the SI pump is 1.37E-3/hr, then the conditional probability table associated with the node *Low P SI* is shown in Table 7.2. Note that if the pump were ON in the current

time step, we assume that it will remain ON in the next time step. If this were not the case, then the last row in Table 7.2 will need to be adjusted accordingly.

Table 7.2. Conditional probability table for the SI pump, assuming a failure probability of 1.37E-3 during 1 hour of operation.

	Time *T*	
Time *T-Δ*	OFF	ON
OFF	1	0
ON	1.37E-3	0.99863

The root nodes representing the failure time of the auxiliary feedwater system (*AFW Fail time)*, the trip time of the primary coolant pump (*RCP Trip Time)*, the time delay between the loss of feedwater and the reactor scram (*Scram Delay*), and the F&B initiation delay (*Feed Delay* and *Bleed Delay*) are all constant across different time slices. This is intuitive since these variables do not change across time slice. Note that these constant nodes appear under the black line in Figure 7.1 to indicate that they are not specifically part of any one slice but are shared among all slices.

In the model, we decided to allow the nodes *No. PORV Opened*, *Low P SI pump* and *No. Hi P SI pump* to vary with each time slice. Physically, this means that the number of pumps and PORVs that are operational may change with time slice. The change in the state of these nodes between different time slices may be caused by stochastic failures, or by deterministic automatic and manual controls. For instance, if we are interested in including human behavior modeling into the network, we may treat the manual control as stochastic to account for variation in operator response time. Human behavior is not considered in this example. To complete the Bayesian network model, we need to specify the root node probability or probability distribution. Table 7.3 shows the values that we

use in the model. The choice of the root node and their probabilities are based on [She11], although some numbers are adjusted for our model. We will next describe the significance of some of these variables.

In normal circumstance, the reactor is automatically set to scram when the water level in the steam generator falls to about 30% of the narrow range indicator. In the Zion-1 plant, this occurs approximately 30 s after the loss of main feedwater if there is no auxiliary feedwater flow. There are situations where the operator may be aware of the feedwater loss before the steam generator water level drops to the automatic trip setting and in these cases, the reactor may be manually tripped before automatic scram occurs. It is also possible that the loss of main feedwater is not abrupt, leading to a longer delay between the start of the loss-of-feedwater event and the steam generator low level signal. To account for these possible scenarios, we model the reactor scram delay time as a uniform distribution between 0 and 90 s.

Table 7.3. Adjustable parameters in the Bayesian network model.

Description	Probability Distribution
Reactor scram delay	Uniform between 0 and 90 s
F&B initiation delay	Lognormal with mean = 4 mins and sigma = 2 mins
Number of SI pump available	Failure prob=0.012/demand per pump, Failure rate=1.37E-3/hr
Number of charging pump available	Failure prob=0.0078/demand per pump, Failure rate=1.37E-3/hr
Number of PORV opened	Failure to open prob=0.014 per valve
Delay between feed initiation and bleed initiation	Normal with mean=20s and sigma=10s
Time of RCP trip	Not tripped
AFW failure time	Failure prob on demand=0.9. Probability of successful actuation but fails <30 mins=0.1

Another variable that will affect the core temperature is the delay between the loss of feedwater and the initiation of the F&B procedure. As can be seen in Figure 5.3, following the loss of main feedwater, the plant operator has to go through a series of procedural steps before F&B operations are initiated. These steps, including verification of the loss of feedwater and implementing the various methods to manually activate the AFW system, will cause some delay in the initiation of the primary feed. Since the exact timing that the operator will enter the F&B emergency operating procedure will depend on the plant state and indicators in the control room, there is a large distribution in the time delay before the feed is initiated. Following the recommendation from [She11], we will model the time to initiate F&B distribution as a log-normal distribution, with a mean of 4 mins.

After the F&B operation is implemented, the bleed rate is another important parameter that will affect the time evolution of the primary coolant temperature. If the bleed rate is too low, the primary coolant pressure will increase, leading to a decrease in the injection capacity and heat removal from the fuel. Conversely, a bleed rate that is higher than the injection rate will lead to a loss of coolant scenario. In our analysis, the coolant bleed rate from the primary system is determined mainly by the number of PORVs that are opened. We assume that the probability that any one of the PORVs will fail to open is 0.014. Note that in addition to the PORVs, the pressurizer also has safety valves that will automatically open on high pressurizer pressure. These safety valves are not normally used in the F&B operation, although there can be situations when the relief capacity of the PORVs is too low to relieve the primary system pressure. In these situations, the safety valves will open and provide additional relief paths.

Generally, for analysis of the feed-and-bleed scenario, we are interested in situations where the AFW system does not actuate. In an operating power plant, after the loss of main feedwater, the operator will try to manually start the AFW system if it does not come on automatically. The entry into the emergency operating procedure which prescribes the F&B operation will not occur unless attempts to start the AFW fail. Nevertheless, data from the NUREG/CR-6928 [NRC07] indicate that even in situations where the AFW system successfully operates, the F&B operation may still be necessary if the auxiliary pumps fail before the core cools down sufficiently. In an attempt to account for this scenario, we will assume that there is a 10% chance that the AFW system will successfully actuate on demand, but will fail within 30 minutes of the actuation.

One last parameter that we should mention is the reactor coolant pump (RCP) trip delay. Generally, the coolant pump is not tripped until the core power is below a certain fraction of the rated power (usually 1%). The trip is initiated to reduce a source of heat addition to the primary coolant water. However, in scenarios such as a station blackout, the coolant pump trip will coincide with the loss of feedwater. In most published studies of the feed-and-bleed scenario, the coolant pump is tripped at the beginning of the transient to ensure that the results are conservative. However, in our analysis, we will not trip the pump in most of the cases presented in Section 7.2. This decision means that the feed-and-bleed procedure in our model will cool the core more effectively than would be suggested in other studies.

7.2 RELAP5 Results for Representative Cases

In this section, we will present the behavior of the reactor after the loss of main feedwater as obtained from RELAP5 calculations. The results obtained here will be used to construct the code surrogates from which we can construct the deterministic conditional "probability" function needed to solve the Bayesian network. The surrogates will also allow us to perform a Monte Carlo sampling and obtain the probability distribution of the non-root nodes much more efficiently than through repetitive RELAP5 runs.

In our RELAP5 model, the location of the maximum clad temperature occurs at node 5 of volume 335 (see Figure 6.1). The initial sequence of events for all the cases in this section is based on the reference sequence shown in Table 7.4. Any deviations from this baseline case will be noted in the description of the individual cases. Following the loss-of-main feedwater at 0 s and the associated failure of the AFW system to engage, the reactor will automatically scram when narrow range water level in the steam generator reaches 25%. For the reference case, the scram time is assumed to be 30s.

Table 7.4. Sequence of event following the loss of main feedwater for the reference case. The RCP is not tripped in this scenario.

Event	Time [s]
Loss of Main Feedwater	0 s
Failure of Auxiliary Feedwater System	0 s
Reactor Trip	30 s
Turbine Valve Closed	30 s

A plot of the fuel cladding temperature as a function of transient time is shown in Figure 7.3 for the baseline case where F&B is not initiated. For all scram delay times up to 90 s, the clad temperature remains relatively constant until the secondary water inventory in the steam generator is depleted. The time of depletion varies with the scram

delay. With a 0 s delay (scram is coincident with the loss of feedwater), the steam generator dryout occurs approximately 90 minutes into the transient. With a 90 s delay, the dryout occurs much earlier at around 30 minutes. After the dryout occurs, the clad temperature rises rapidly for all delay times. The rate of temperature increase after the steam generator dryout is relatively independent of the scram delay. As expected, the case of a 90 s scram delay leads to the highest clad temperature of 1800 ^{o}F after 4 hours.

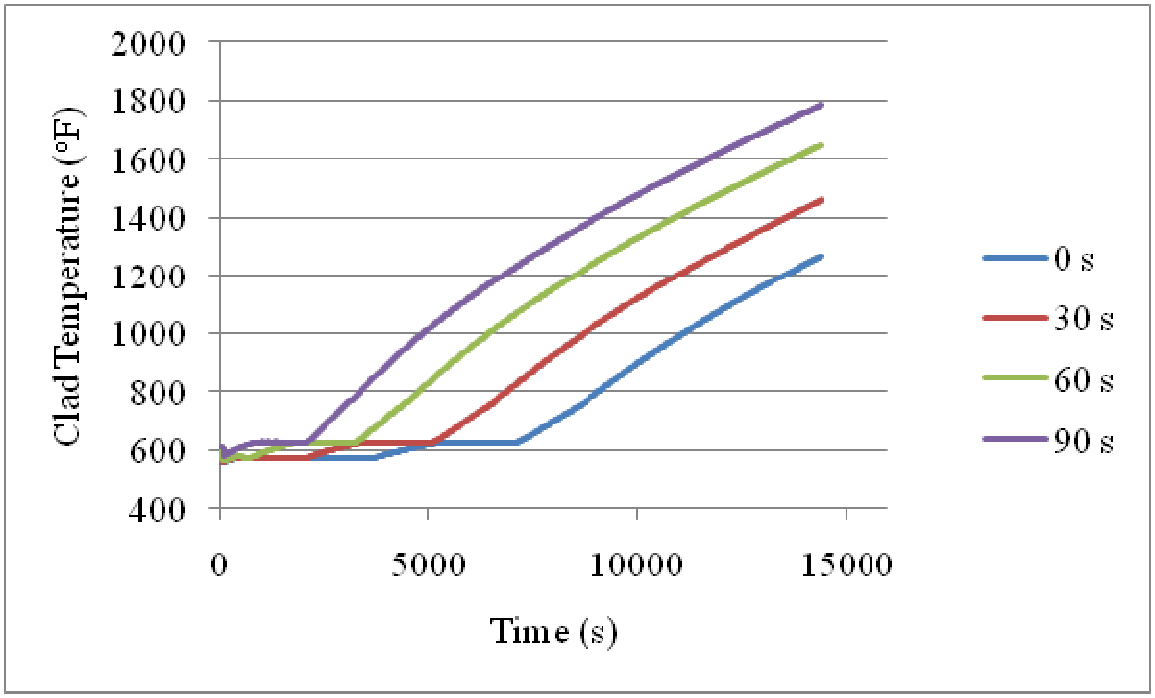

Figure 7.3. Peak clad temperature as a function of transient time for the scram delay time of 0 s, 30 s, 60 s, and 90 s. F&B is not initiated.

Figures 7.4 and 7.5 show the variation of the average coolant temperature and pressure as a function of transient time. From the plot, the primary pressure remains essentially constant until the F&B operation is initiated at around 7200s. The opening of the PORVs associated with this operation causes the pressure to drop by over 50%. The coolant temperature rises steadily after the LOFW until the time of the F&B operation. At 7200 s, the temperature drops sharply as an extra heat removal pathway is established.

The dependence of the clad temperature on the scram delay is weaker in the case where the F&B operation is initiated early in the transient. Figure 7.6 shows the case where the F&B operation is initiated 10 minutes after the loss of feedwater. The clad temperature drops initially following the scram, then briefly rises over the next 600 s before the F&B operation is initiated. Once the feed flow starts, the temperature begins to fall at a rate that is almost independent of the scram delay time. A steam generator dryout does not occur in this scenario since the F&B operation provides the main path of decay heat removal.

In Figure 7.7, we see the effect of the time delay for the initiation of F&B on the clad temperature. With a delay of 10 minutes, the clad temperature does not rise much from the initial value, and it falls at a constant rate once the primary feed starts. As expected, the peak for the clad temperature during the transient is highest in the case where the F&B operation is delayed the longest.

The effect of the number of PORVs that are successfully opened for the bleed operation is shown in Figure 7.8. The rate of cooling provided by the F&B operation is lower for the case where one of the PORVs failed to open. The lower bleed flow rate results in a higher primary pressure which in turn results in lower feed rate (since the feed rate is pressure dependent). The effect of primary system pressure on the total feed rate is shown in Figure 7.9. At pressures near the steady state operating pressure of 2252 psia, only the charging pump has enough power to inject water into the core. Fig 7.10 shows the effect of feed flow rate on the clad temperature. As expected, cooling is achieved at a faster rate in situations where the feed rate is high.

Most studies on the viability of the F&B operation use a conservative assumption that the AFW system fails to initiate immediately following the loss of feedwater. However, if the auxiliary system does actuate but fails early into the transient, the impact on fuel temperature will be similar to the cases described in the preceding paragraphs. Figure 7.11 looks at how a delayed AFW system failure will affect the scenario. The plot suggests that an AFW system failure delay merely prolongs the initial period where the clad temperature remains relatively constant. After the failure, the behavior is similar to all the previous cases. Thus, the AFW delay merely shifts the clad temperature profile to a later point in the transient.

In this section, we looked at how some of the parameters listed in Table 7.3 impact the time evolution of the peak clad temperature. The results shown in this section all assume that only one parameter changes while all the other parameters remain constant. However, if we need to either perform any type of sampling for a Monte Carlo simulation or to construct a conditional probability function for the Bayesian network, we need to be able to predict how the clad temperature responds, given any random values of the independent variables. To achieve this goal, we will need to use the conditional independence assumptions that were made when constructing the Bayesian network model in Figure 7.1. The next step in building the conditional probability function will be to develop surrogates for all the RELAP5 cases so that we can avoid the need to run the RELAP5 case for every parametric case. These surrogates will be discussed in the next section.

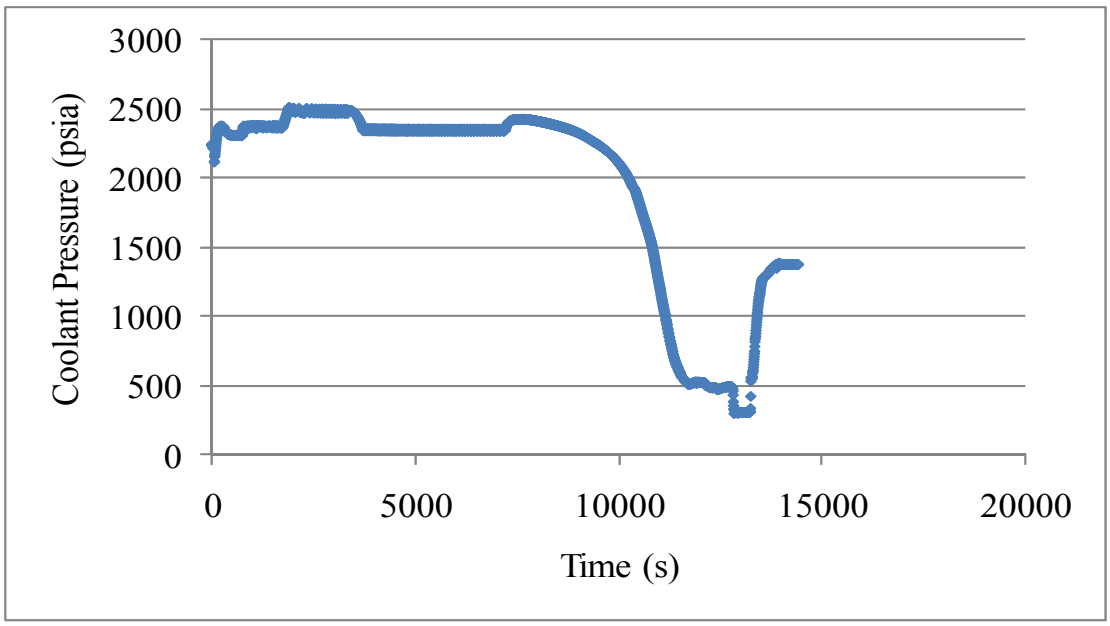

Figure 7.4. Primary coolant pressure as a function of time for 30 s scram delay and 2 hours F&B initiation delay.

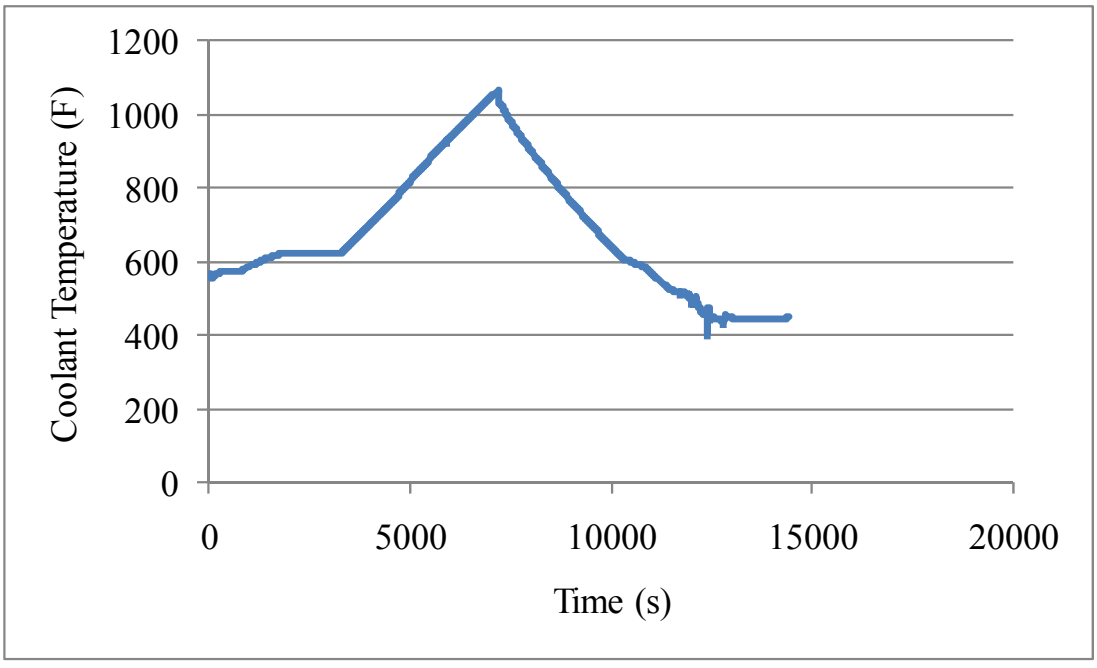

Figure 7.5. Primary coolant temperature as a function of time for 30 s scram delay and 2 hours F&B initiation delay.

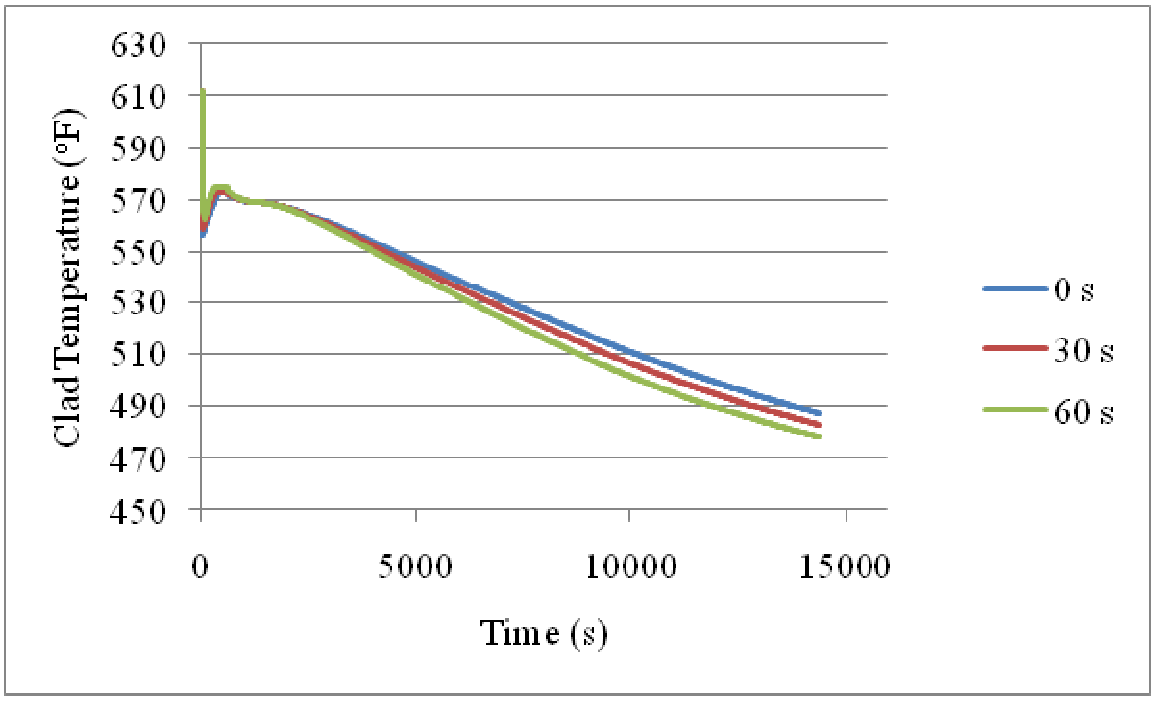

Figure 7.6. Peak clad temperature as a function of transient time for the scram delay time of 0 s, 30 s, and 60 s. F&B is initiated 10 mins after the LOFW.

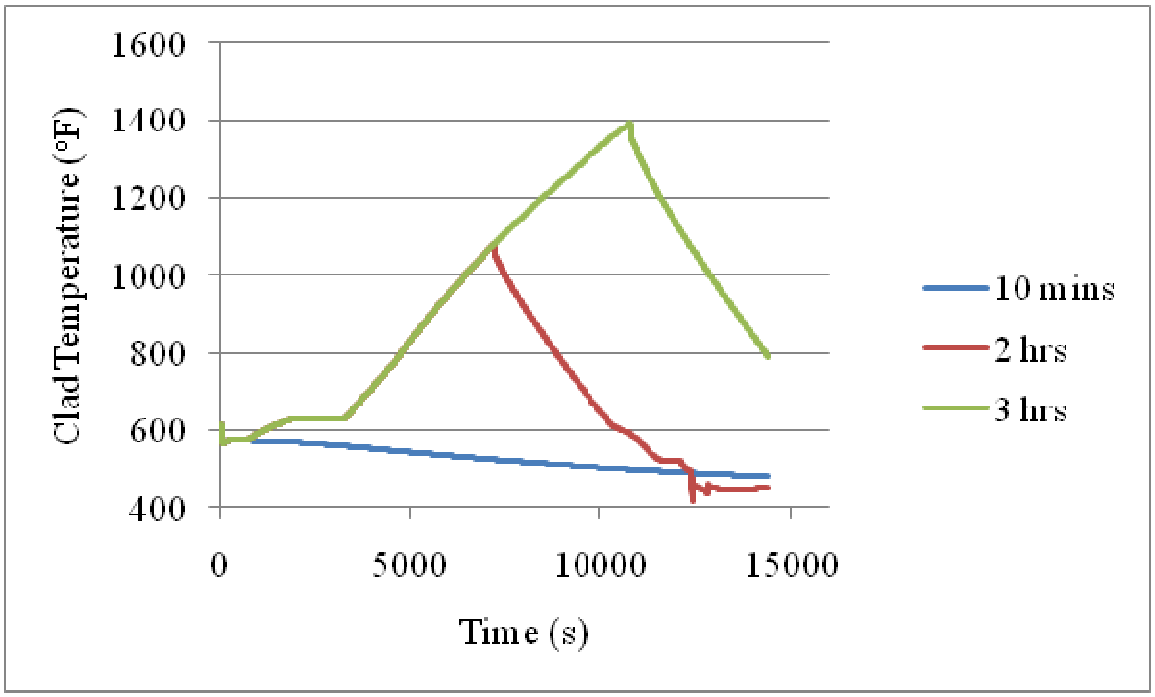

Figure 7.7. Peak clad temperature as a function of transient time for the F&B initiation delay of 10 mins, 2 hrs, and 3 hrs after the loss of feedwater. The scram delay time is 60 s for all three cases.

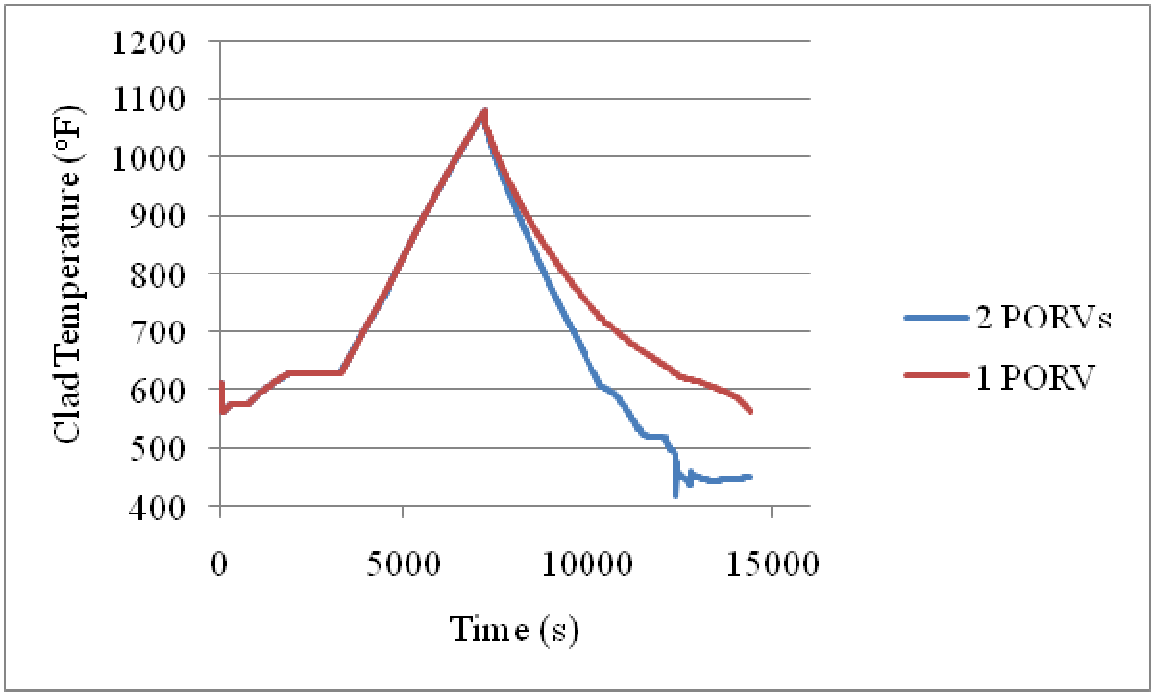

Figure 7.8. Peak clad temperature as a function of transient time for cases where 1 and 2 PORVs successfully opened. The scram delay time is 60 s and the F&B operation is initiated 2 hours after the loss of feedwater.

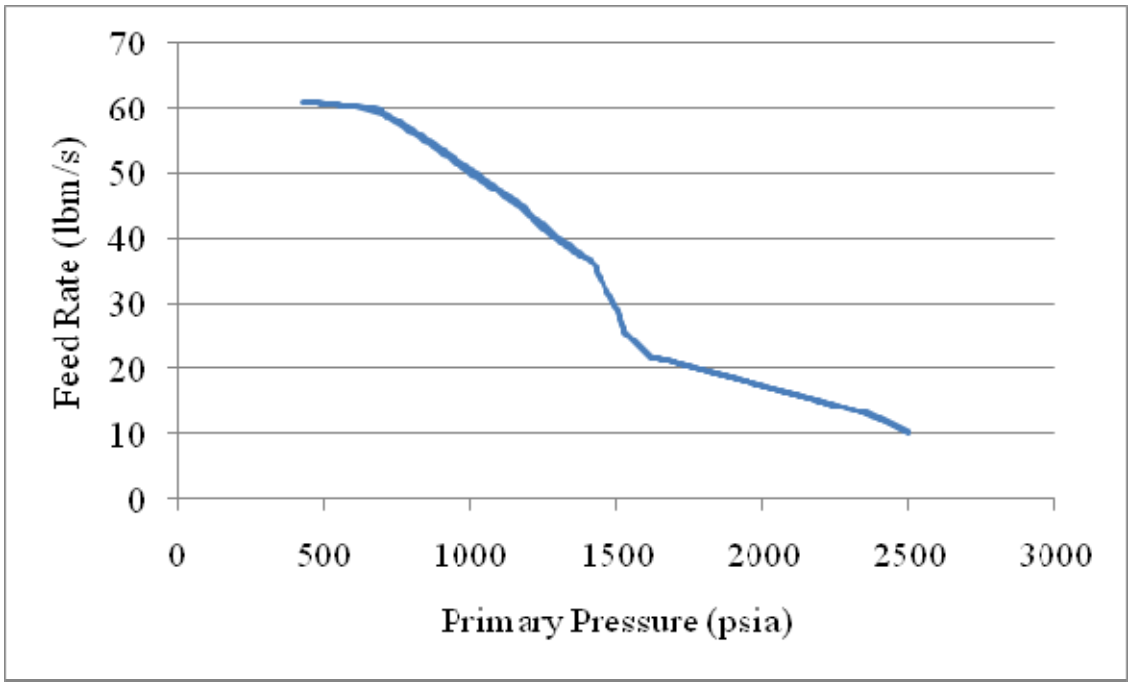

Figure 7.9. Feed flow rate (from both high and low pressure injection systems) as a function of primary pressure.

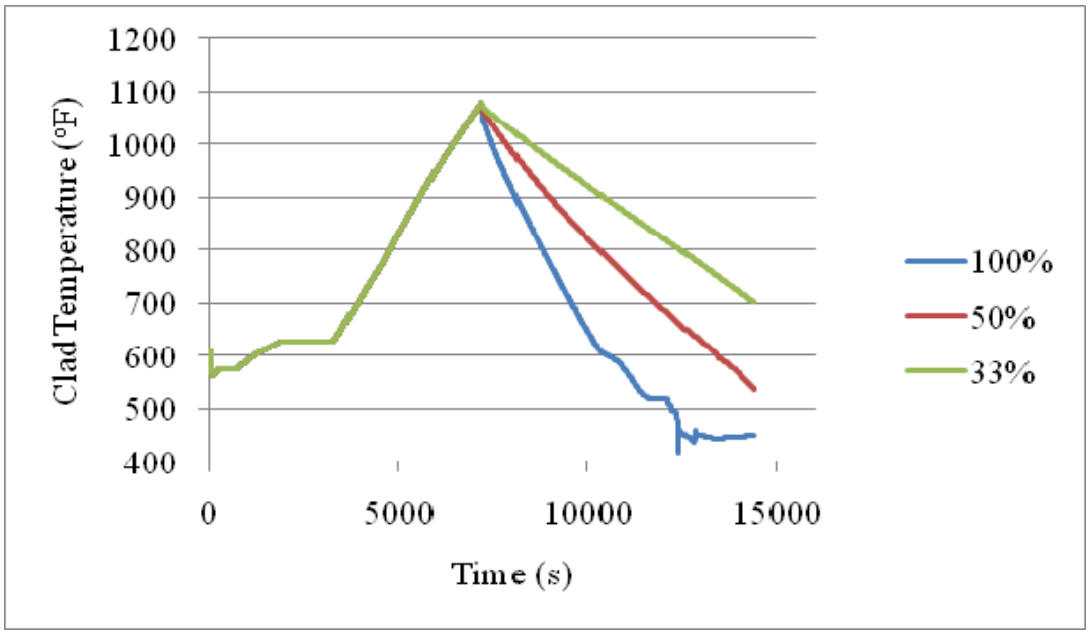

Figure 7.10. Peak clad temperature as a function of transient time for cases where 33%, 50%, and 100% of the makeup flow (from both the high and low pressure injection systems) are available. The scram delay time is 60 s and the F&B operation is initiated 2 hours after the loss of feedwater.

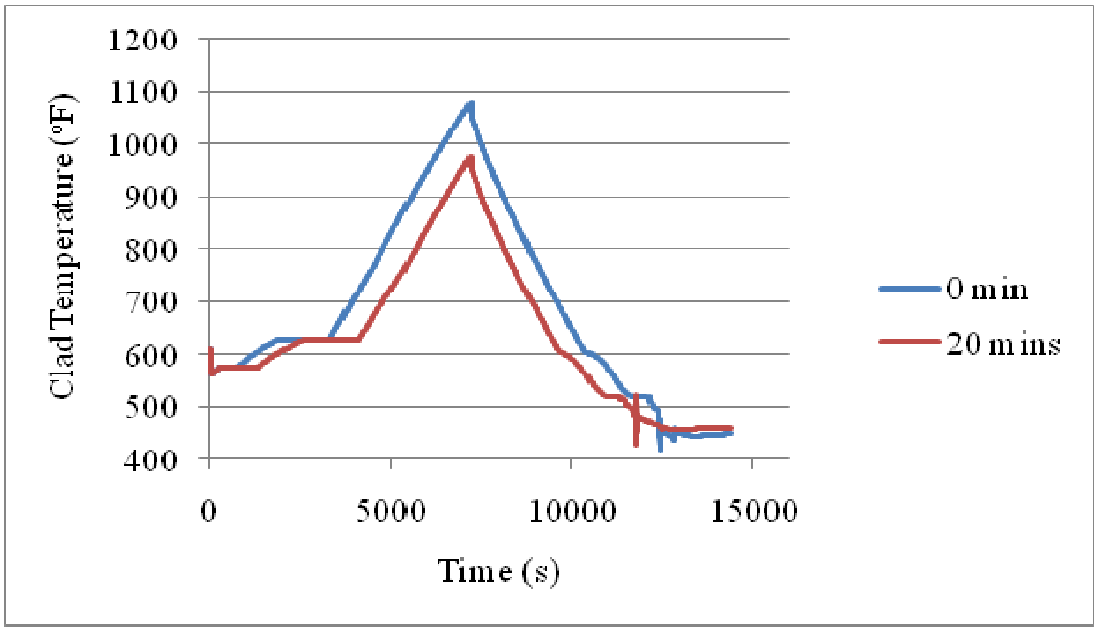

Figure 7.11. Peak clad temperature as a function of time for the case where the AFW fails immediately and 20 mins after the loss of feedwater. The scram delay time is 60s and the F&B operation is initiated 2 hours after the loss of feedwater.

7.3 Constructing the RELAP5 Surrogates using ACE

Referring to Figure 7.1, we wish to find the relationship between the clad temperature (dependent variable), and the primary coolant pressure, primary coolant temperature, and the coolant flow rate (three independent variables). We compile the RELAP5 cases that were run varying these three independent variables (about 20 cases) and tabulate the functional behavior of the clad temperature on changes in the primary coolant pressure, temperature, and flow rate. This procedure generates about 5000 data points which give the relationship between the four variables. The matrix containing these data is then used in the ACE algorithm to generate the transformed variables. Figure 7.14 shows the result of the transformation. The $R^2 = 0.997$ value for the fit suggests that the linear additive relationship

$$\theta(Y) = \phi_1(X_1) + \phi_2(X_2) + \phi_3(X_3) \tag{7.1}$$

is valid.

Earlier in this chapter, we mentioned how some parent node may contribute less to the behavior of a child node compared to other parents. From Figure 7.14, we see that the variable X_2 (*Primary Water Pressure*) contributes less than 1% of X_1, and 10% of X_3 to the transformed dependent variable $\theta(Y)$. This means that the coolant temperature and hence the clad temperature is relatively independent of the primary coolant pressure, at least within the scope of our model. However, for the sake of completeness, we will keep *Primary Coolant Pressure* as a parent of *Primary Coolant Temperature*.

Next, we need to determine the functional behavior of the primary coolant temperature on its dependent variables as specified in Figure 7.1. Using the same step that we used to develop Figure 7.14, we obtain the transformations shown in Figures 7.15

and 7.16. The R^2 value associated with the surrogate is 0.979. Note that unlike the case of the clad temperature model in Figure 7.10, the primary coolant pressure is the dominant parent in determining the state of the coolant temperature. The relationship between the transformed water temperature and the untransformed temperature is shown in Figure 7.16. Note that we do not need a RELAP5 surrogate for the *Injection Flow Rate* node. The dependence of the feed flow on the number of pumps operating and the primary system pressure can be taken directly from Table 7.1.

Figures 7.12 and 7.13 show scattered plots of the clad temperature and coolant temperature versus their respective independent variables. The data from these plots were obtained by combining the 50 RELAP5 cases that were used to generate the surrogates. From these plots, we expect to see a mostly linear relationship between clad temperature and coolant temperature, but the relationship of the other variables are less clear. Note that the multi-valued nature of these plots suggest that each of the independent variables are not sufficient to determine the dependent variable by itself. The surrogates obtained from Figures 7.12 and 7.13 are shown in Figure 7.14 and 7.15. As expected, we see that the relationship between the clad and coolant temperatures are mostly linear. The relationship for the other variables are more well defined in the transformed function space compared to the original data.

Figures 7.17 through 7.19 show the benchmarking of the surrogates for the coolant pressure, clad temperature, and coolant temperature against the values obtained from RELAP5.

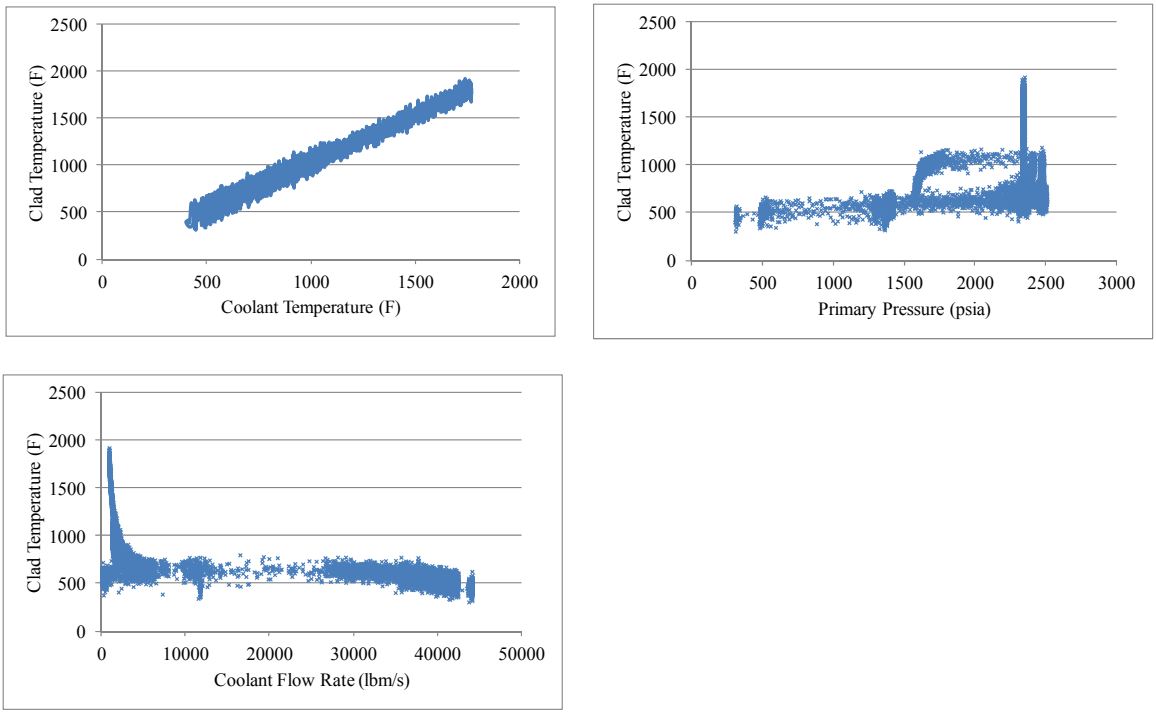

Figure 7.12. Scatter plots of clad temperature as a function of its independent variables.

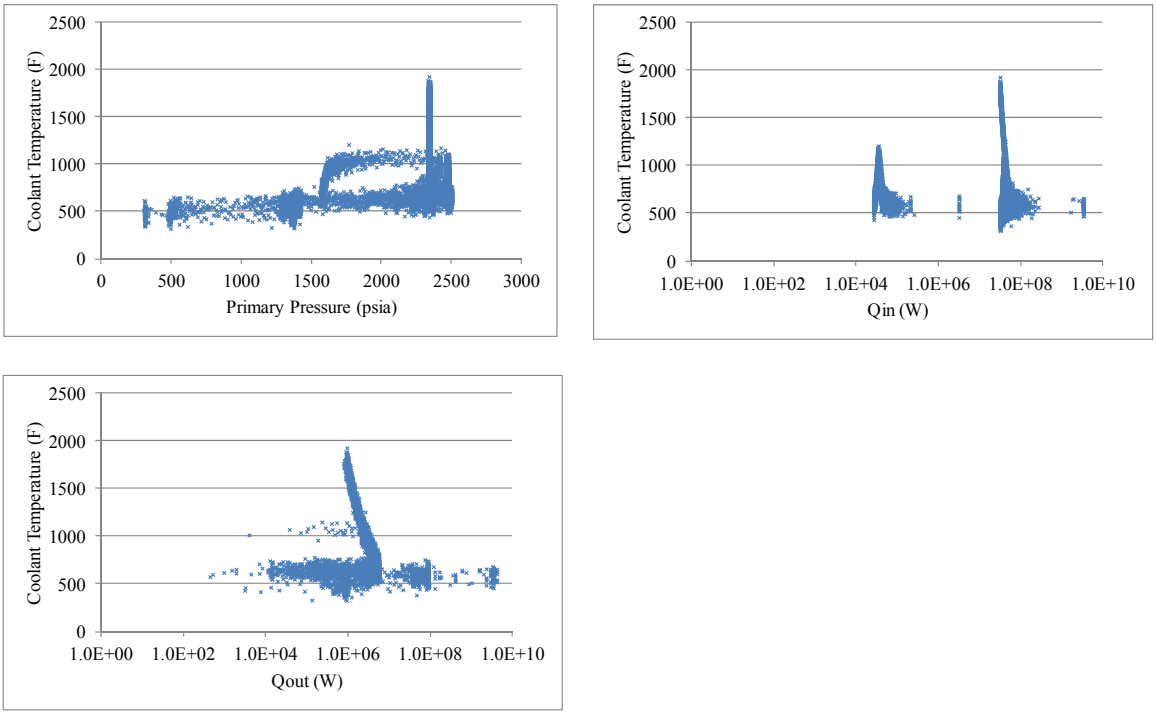

Figure 7.13. Scatter plots of coolant temperature as a function of its independent variables.

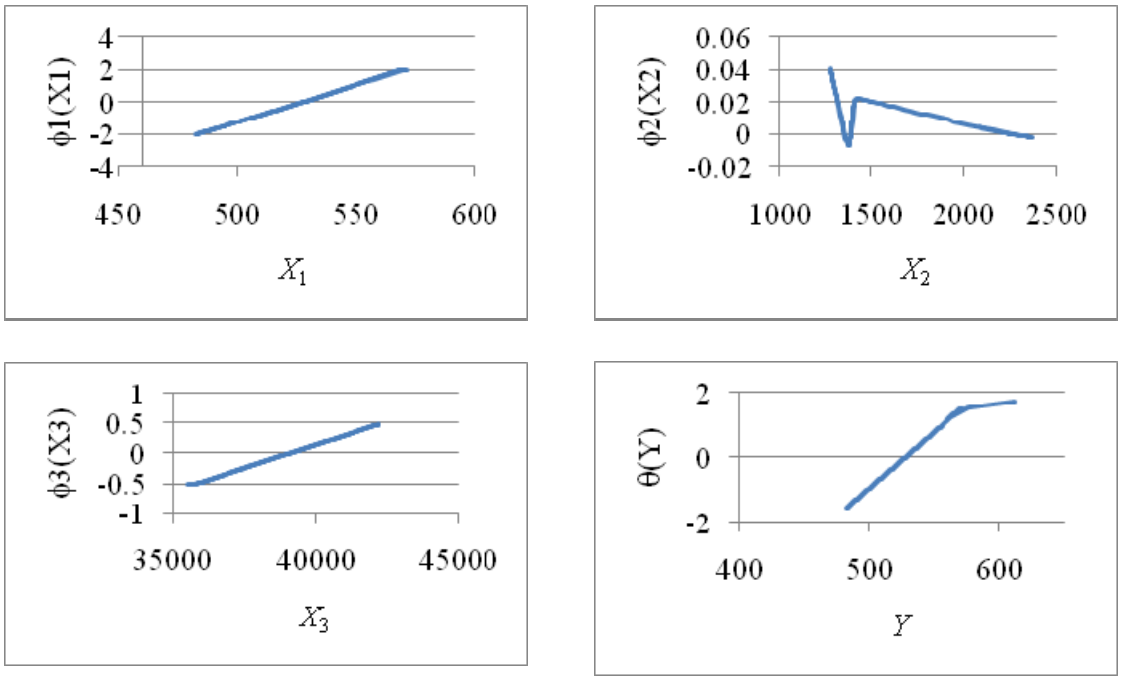

Figure 7.14. Clad temperature as a function of three transformed variables obtained via the ACE algorithm. X_1, X_2, X_3 are the coolant temperature, pressure, and flow rate, respectively. The dependent variable is Y, the clad temperature.

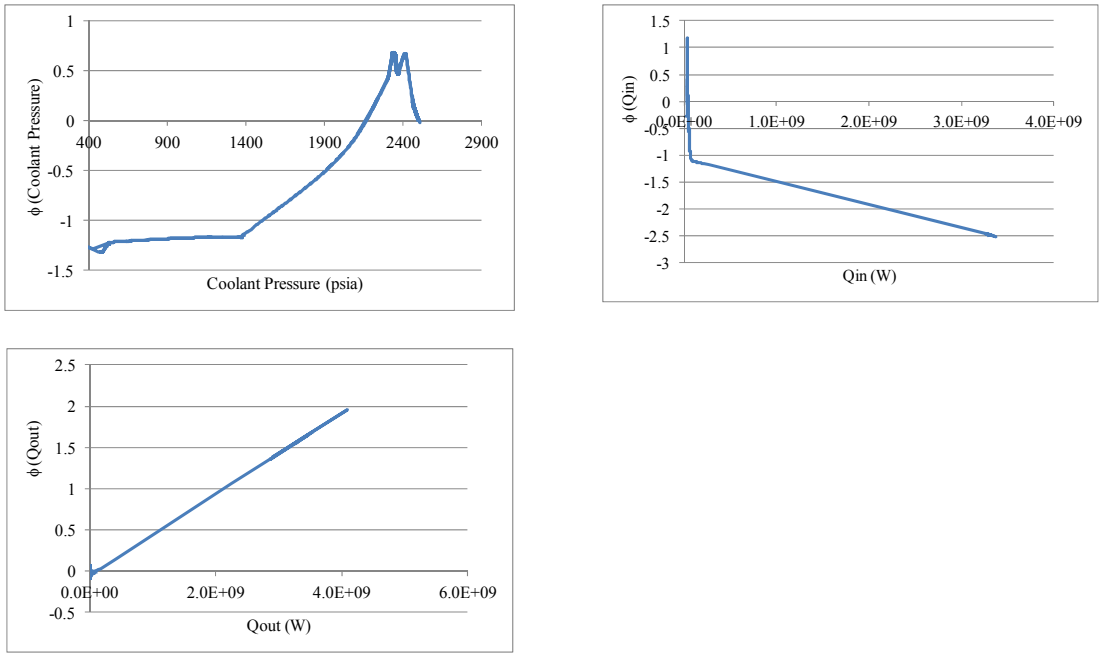

Figure 7.15. Transformed dependent variables as a function of the original variable for the *Primary Coolant Temperature* node.

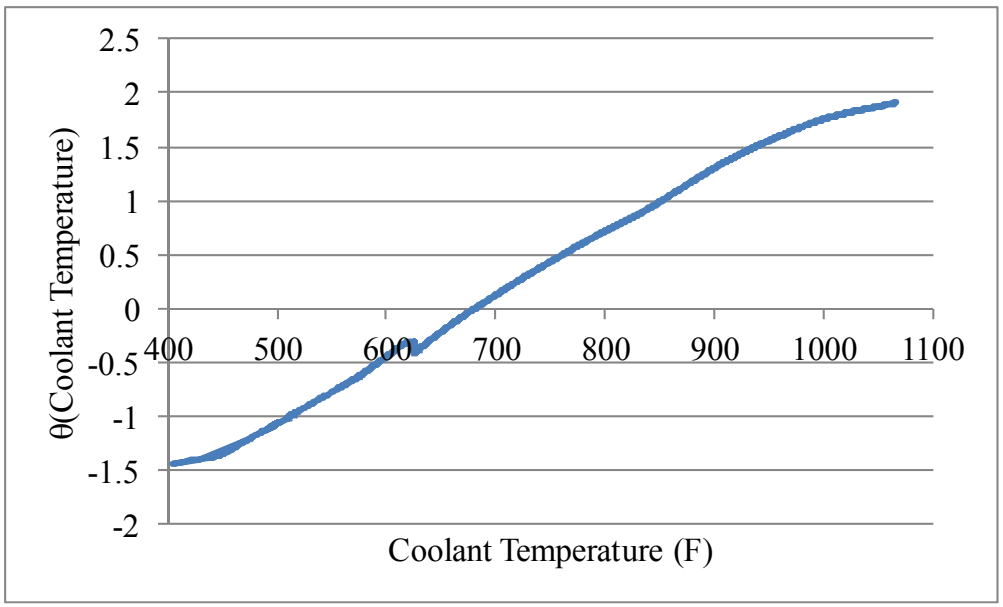

Figure 7.16. Transformed primary coolant temperature as a function of the original variable.

We note that although we used a quasi-static state assumption in the DBN structure of Figure 7.1 to generate these surrogates, we are able to reproduce the time-dependent behavior of the plant variables. Generally, we may not expect that a quasi-state state assumption will allow the surrogates to track the process variables well during periods of large changes (for instance, the large drop in pressure after opening of the PORVs). However, Figures 7.17 - 7.19 show that even when there are large variations in the variables, the surrogate still perform adequately. The values of the independent variables that are used in the benchmarking in the above figures cover the entire range covered in the sampling. For instance, the scram delay varies from 30 s to 90 s and the F&B initiation delay ranges from 10 mins to infinity (no F&B operation). Therefore, we are relatively confident that the surrogates provide a good approximation for RELAP5, at least in the interval of interest.

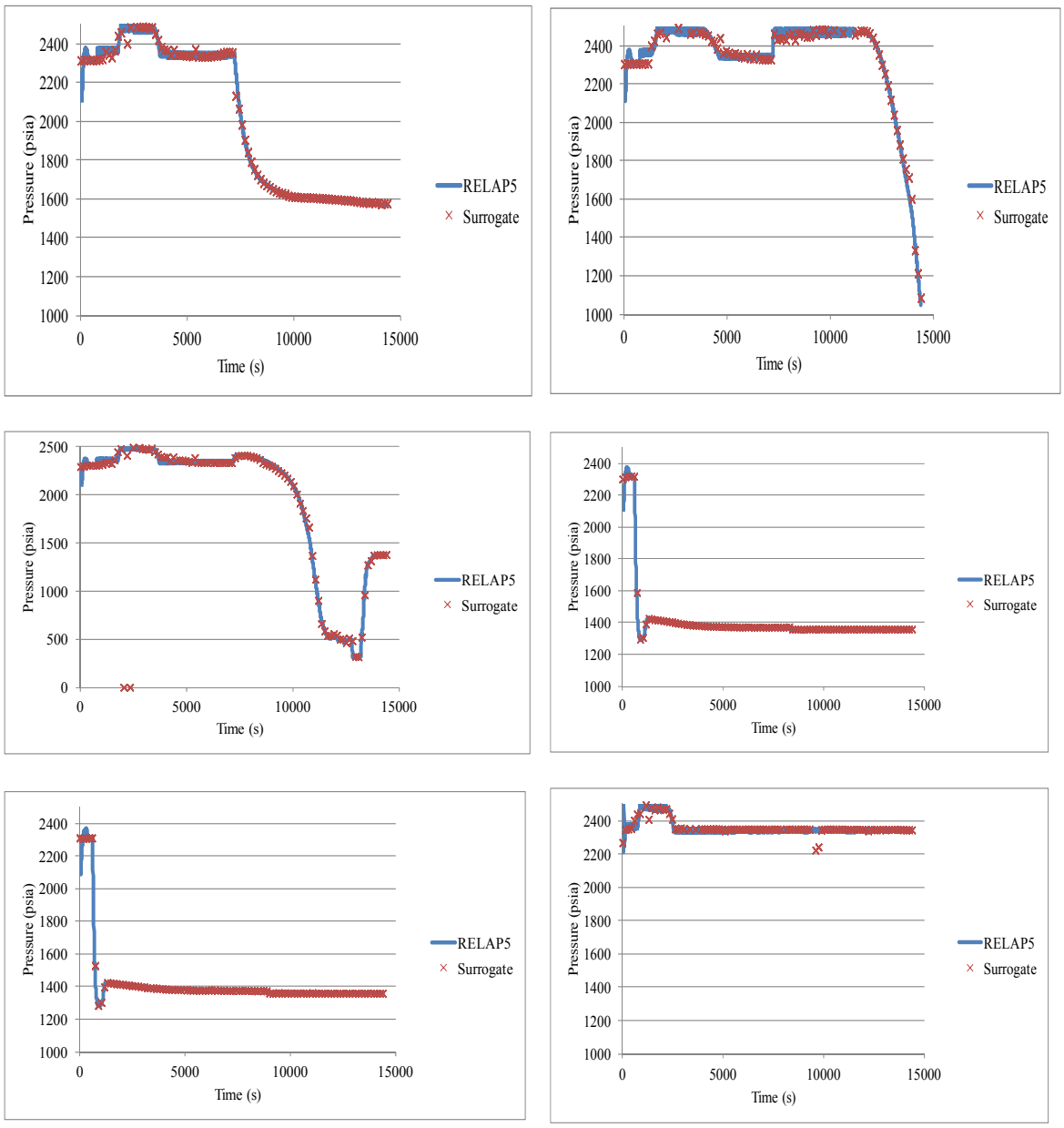

Figure 7.17. Benchmarking of the coolant pressure surrogate. From left to right, the cases are: 33% feed rate (60 s scram delay and 2 hrs F&B delay), 50% bleed rate (60 s scram delay and 2 hrs F&B delay), 100% feed and bleed rates (60 s scram delay and 2 hrs F&B delay), 10 mins F&B delay (60 s scram delay), 10 mins F&B delay (30 s scram delay), and 90 s scram delay without F&B.

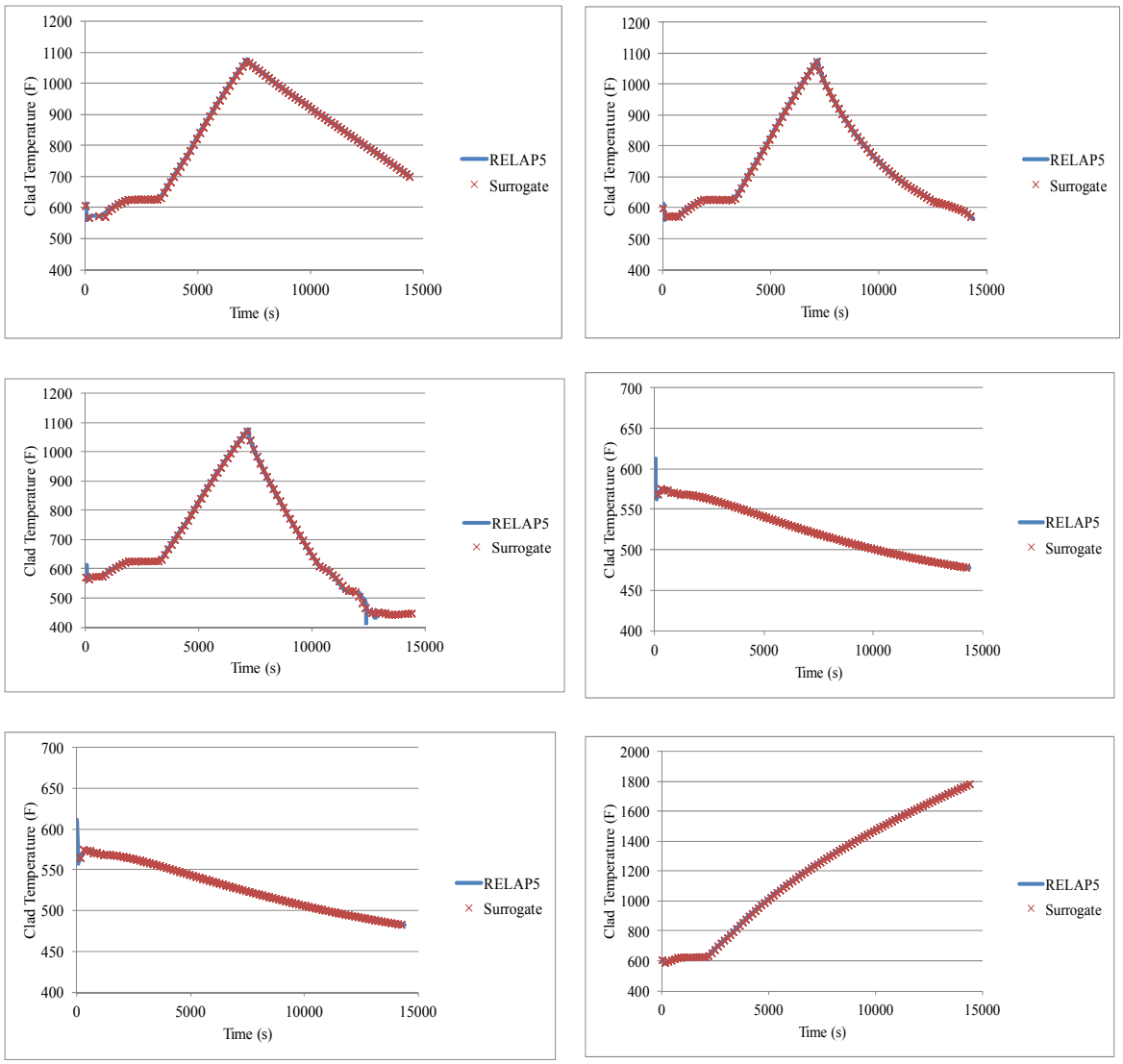

Figure 7.18. Benchmarking of the clad temperature surrogate. From left to right, the cases are: 33% feed rate (60 s scram delay and 2 hrs F&B delay), 50% bleed rate (60 s scram delay and 2 hrs F&B delay), 100% feed and bleed rates (60 s scram delay and 2 hrs F&B delay), 10 mins F&B delay (60 s scram delay), 10 mins F&B delay (30 s scram delay), and 90 s scram delay without F&B.

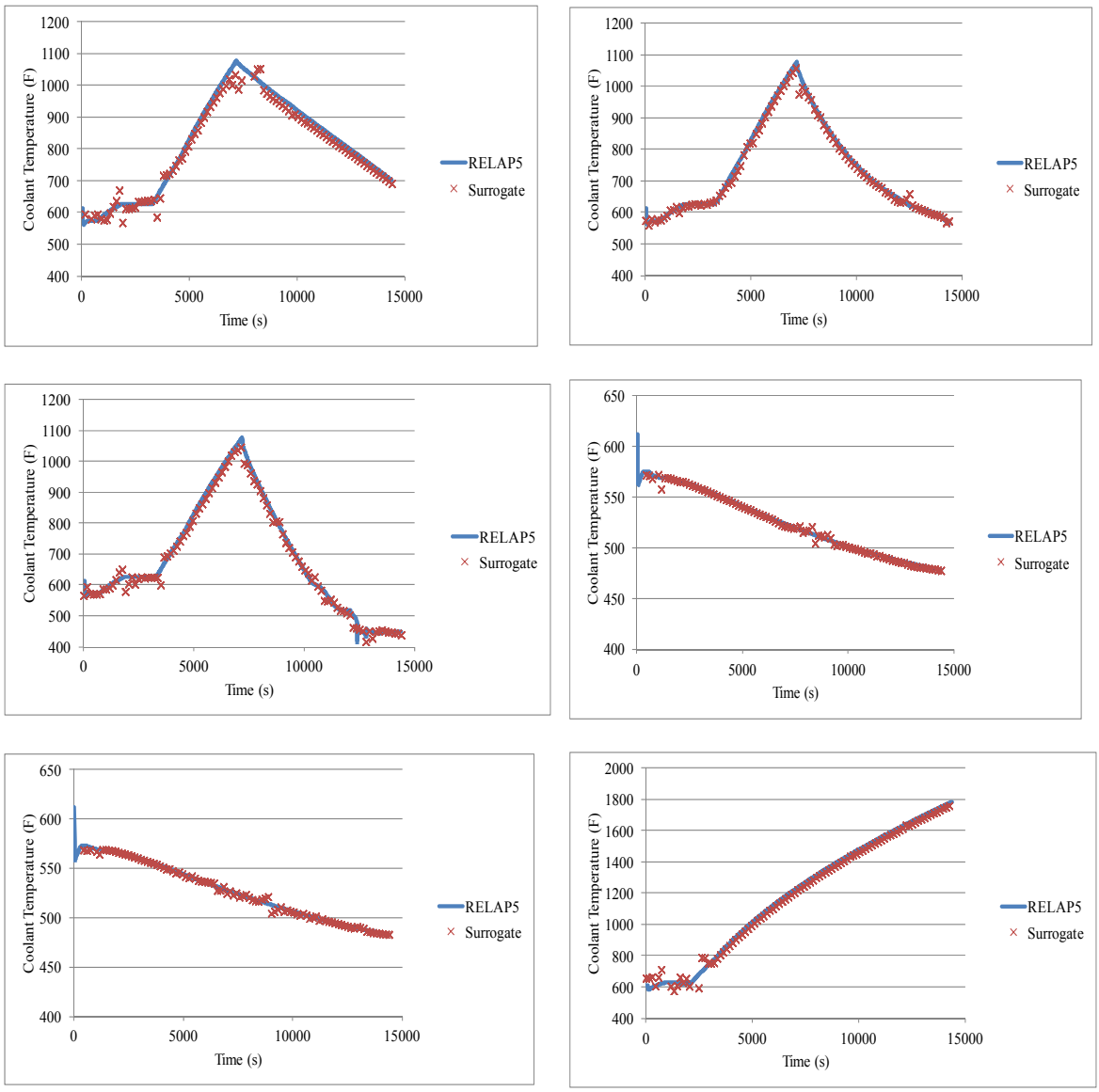

Figure 7.19. Benchmarking of the coolant temperature surrogate. From left to right, the cases are: 33% feed rate (60 s scram delay and 2 hrs F&B delay), 50% bleed rate (60 s scram delay and 2 hrs F&B delay), 100% feed and bleed rates (60 s scram delay and 2 hrs F&B delay), 10 mins F&B delay (60 s scram delay), 10 mins F&B delay (30 s scram delay), and 90 s scram delay without F&B.

7.4 Risk Analysis Result for the Loss of Feedwater Transient

With the data presented in the previous two sections, we now have sufficient information to calculate the core damage probability given the root event probability distribution in Table 7.3. This distribution is generated by weighting the distribution of the fuel clad

temperature that we obtained with the Monte Carlo sampling of the ACE surrogates for the probability of core damage. The plot, shown in Figure 7.20, represents the probability of having core damage at each time step. At 15000 s into the transient, the core damage probability peaks at 5E-3. From Figure 7.7, we can see that the peak clad temperature can occur anywhere from 10 mins, in the case of early F&B initiation, to over 3 hours if the F&B operation is delayed. In all of these cases, the clad damage probability will either increase or remain constant after the peak temperature is reached.

Looking at the root node probabilities, we expect that the dominant contributions to clad failures will come from cases where both the scram delay and the F&B initiation delay are large. Since we use a uniform distribution for the scram delay time, the Monte Carlo history which has a large delay time occurs relatively frequently. The F&B actuation delay largely determines when the peak clad temperature will occur. After the F&B operation is initiated, the clad temperature falls rapidly, leading to a near constant fuel damage probability.

The Bayesian network is solved using 80,000 Monte Carlo samples from the distributions given in Table 7.3 . Figure 7.20 shows the time-dependent failure probability for the LOFW transient. The plot indicates that about 3.5 hours into the transient, we can expect approximately a 0.5% probability of core damage. Unlike the event tree method, it is difficult to determine the cut set or the relative contributions of different sequences to the final core damage probability via the Bayesian network method. Nevertheless, we can make an educated guess on the qualitative behavior of the plot. The rise in the damage probability after 3.5 hours is likely a result of a combination of high scram delay time, low injection rate, and the long F&B initiation delay.

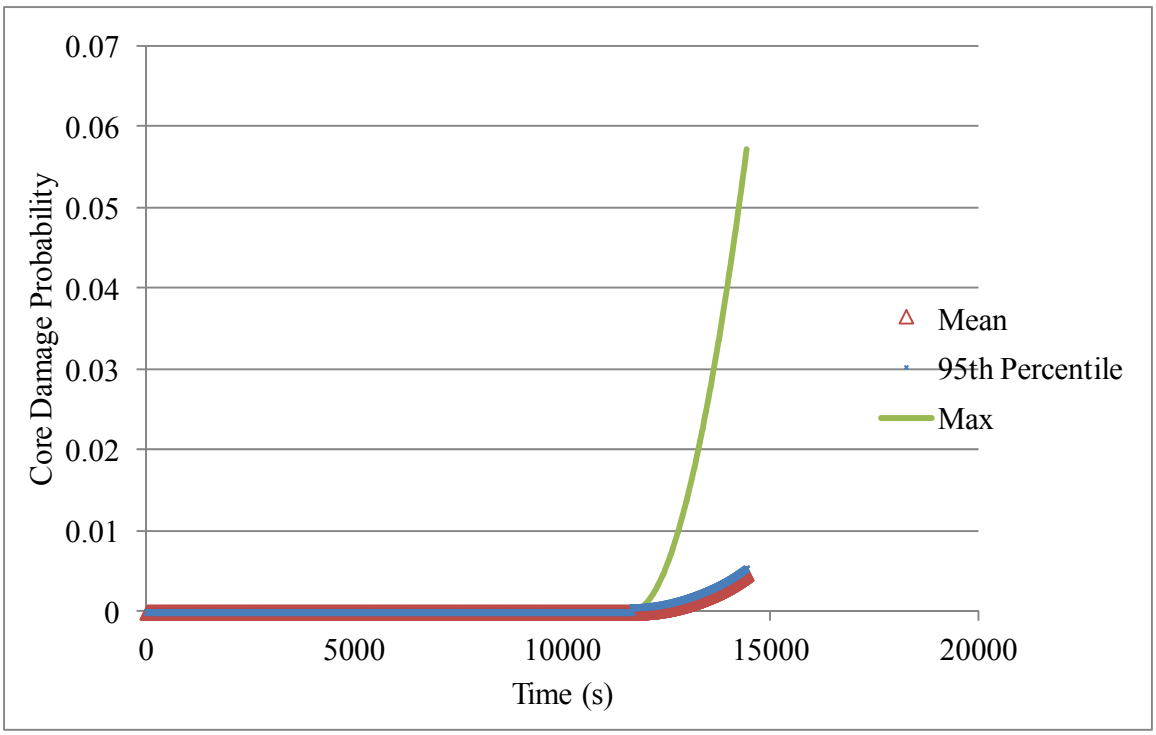

Figure 7.20. Core damage probability as a function of transient time. The Max plot is the result of the one case with no F&B initiation and 90 s scram delay.

In Fig. 7.20, we have also included the core damage probability associated with the case of no F&B operation and with a 90 s scram delay. This case gives the highest clad temperature at any point in time among at the cases that are included in the simulation. Therefore, this case serves as the upper bound for our core damage probability.

In our analysis, we have shifted the threshold of core damage downward by approximately 200 ^{o}F. This was done to allow more of our Monte Carlo samples to have a nonzero probability of core damage so that we obtain adequate number of cases that produce core damage. One way to avoid this modification will be to increase our end time for the analysis of 15000 s. This will allow the clad temperature in more Monte Carlo cases to produce core damage. The narrowing of the density function in Figure 7.2

will result in a shift of the time for the onset of core damage to the right. In Figure 7.20, this will translate to a faster rise in core damage probability after about 3.5 hours. We also expect the damage probability to increase beyond 15000 s and it will be interesting to investigate when the peak probability occurs.

CHAPTER 8

SUMMARY AND CONCLUSIONS

Probabilistic risk analysis methods can generally be divided into two groups: the classical static techniques and the newer dynamic techniques. Static PRA makes certain simplifications regarding the time evolution of the process variables in the reactors and analyzes the effects of different initiation events by using these assumptions. Reactor analysis codes can be employed during the construction of the model to determine the behavior of the reactor under the assumed reactor configuration and how the configuration changes during the accident. Static PRA techniques work well when the accident evolution falls under the range of validity for the simplifying assumptions. However, in situations where the changes in plant state have a large impact on the accident progression, the static methodologies may yield erroneous results. These situations include cases where the operator action may be a strong function of the process history or in cases of standby systems where the exact plant configuration at the time of demand may be variable. In these instances, a dynamic PRA approach is the preferred method.

A brief overview of some commonly known dynamic PRA techniques are discussed in Chapter 2. Most of these techniques have a common drawback that they are quite complicated for applications in large systems. In the dynamic PRA approach, the process variable evolution is incorporated directly into the models, leading to a direct

coupling of risk analysis with plant state changes. However, with the a large number of path that the plant state may follow and the complex response behavior of the accident progression, the calculations involved in applying these techniques are very CPU and memory intensive. Therefore, in most practical applications, these techniques are only used to analyze subsystems of the entire plant where the use of static PRA is inadequate.

With the two extremes of the PRA methodologies mentioned above, there are recognitions that in some situations, the analysts may not need the full power of dynamic PRA methodologies and yet want to incorporate the plant response directly into the analysis. The work presented in this dissertation aims to bridge this gap between static and dynamic techniques. In this dissertation, we present a method where the core damage probability is calculated as a function of time by using a DBN model in conjunction with the RELAP5 code surrogates constructed via the ACE algorithm. The use of the DBN allows the dependencies among the plant components to be broken into smaller models which are easier to analyze than the full model. For each of these small subsets in the DBN, the number of independent variables that need to be considered is greatly reduced. The use of the ACE algorithm now becomes more efficient since only selected variables need to be considered.

A major drawback of our proposed method is its sensitivity to the conditional independence assumption. If we do not consider the correct parent in the network or if we miss important independent variables during the construction of the surrogate, then the results we obtain will give inaccurate results. Furthermore, the use of the surrogate is restricted to analyses that are performed within its range of validity. If a transient takes the reactor along a path that causes the parameters to fall outside the scope of our

surrogate, then the prediction of the network will no longer be accurate. In these cases, our model will not give any indications that something is wrong, and it will be up to the analyst to ensure that all the parameters used in the DBN are within the range of validity.

In this work, we have applied the DBN model to the calculation of the time-dependent core damage probability associated with a LOFW accident. The conditional independence statements of important plant components are used to construct the DBN model which links the clad temperature to independent variables such as the scram delay time and the F&B initiation time. The conditional probability functions associated with the nodes in the DBN are a combination of the deterministic ACE surrogates of the RELAP5 code and probability densities describing the failure rates of various plant components. The use of the ACE surrogates inside the DBN allows the network to be solved much faster than possible if a direct coupling with RELAP5 were made. The results of our calculation show that the core damage probability begins to increase from 0 at about 3 hours into the LOFW transient, corresponding to the time when the peak clad temperature in most Monte Carlo samples exceeds 1600^{o}F. We note that if we allow the RELAP5 runs to exceed 15000 s, then mostly likely the core damage probability will continue to increase.

The work in this dissertation focuses on using the RELAP5 surrogate and the factorization property of the DBN to allow an integration of the system dynamics with a PRA analysis in a simplified manner. The quasi-state ACE surrogates are able to represent dynamic behavior of the system accurately. Further study in this area will involve extending the analysis to beyond 15000 s and looking at how the core damage probability increases beyond this period. Due to a time limitation, we have restricted our

analysis up to 15000 s after the LOFW and we have shifted the temperature at which core damage starts by 200 ^{o}F. Should the transient be allowed to continue past 15000 s, we expect that this modification will no longer be necessary as the clad temperature will exceed the threshold for clad damage in many more Monte Carlo samples.

An aspect of DBN that is not covered in this research is its potential use in diagnostics and real-time monitoring. With the ability of DBN to update both the conditional independence functions and the root node probabilities with real data, a DBN risk model will be useful for monitoring the risk of the reactor in real time. In this type of application, we can envision the linking of all the control room instrumentation readings into the network and continuously update the risk of the reactor in real time. During an accident or unanticipated transients, the operators will have access to real time information on the risk as emergency procedures are implemented. The use of our DBN/ACE model in this way could be one possible area of future research.

REFERENCES

[Aco93] C. Acosta and N. Siu, "Dynamic Event Trees in Accident Sequence Analysis: Application to Steam Generator Tube Rupture," *Reliability Engineering and System Safety*, **41**, 135 (1993).

[Ald87] T. Aldemir, "Computer-Assisted Markov Failure Modeling of Process Control Systems," *IEEE Transactions on Reliability*, **R-36**, 133 (1987).

[Ald92] T. Aldemir, N. O. Siu, A. Mosleh, P. C. Cacciabue, and B. G. Goktepe, *Reliability and Safety Assessment of Dynamic Process Systems*, NATO ASI Series, Springer-Verlag (1992).

[Ald07] T. Aldemir, "Some Measure Theoretic Issues in Probabilistic Dynamics," *Nuclear Science and Engineering*, **155**, 497 (2007).

[And09] J. Andrews, "System Reliability Modelling: The Current Capability and Potential Future Developments," *Journal of Mechanical Engineering Science*, **223**, 2881 (2009).

[Aum93] S. E. Aumeier and J. C. Lee, "A Monte Carlo Technique for System Diagnostics, Monitoring, and Surveillance," *Transactions of the American Nuclear Society*, **69**, 235 (1993).

[Bel96] M. Belhadj and T. Aldemir, "Some Computational Improvements in Process System Reliability and Safety Analysis using Dynamic Methodologies," *Reliability Engineering and System Safety*, **52**, 339 (1996).

[Ben06] A. Ben Salem, A. Muller, and P. Weber, "Dynamic Bayesian Networks in System Reliability Analysis," *Proceedings of the 6th IFAC Sumposium on Fault Detection, Supervision and Safety of Technical Processes*, 481 (2006).

[Bob01] A. Bobbio, L. Portinale, M. Minichino, and E. Ciancamerla, "Improving the Analysis of dependable Systems by Mapping Fault Trees into Bayesian Networks," *Reliability Engineering and System Safety*, **71**, 249 (2001).

[Bou03] M. Bouissou and J. Bon, "A New Formalism that Combines Advantages of Fault Trees and Markov Models: Boolean Logic Driven Markov Processes," *Reliability Engineering and System Safety*, **82**, 149 (2003).

[Bou05a] H. Boudali and J. B. Dugan, "A New Bayesian Network Approach to Solve Dynamic Fault Trees," *Proceedings of the Reliability and Maintainability Symposium* (2005).

[Bou05b] H. Boudali and J. B. Dugan, "A Discrete-time Bayesian Network Reliability Modeling and Analysis Framework," *Reliability Engineering and System Safety*, **87**, 337 (2005).

[Bou05c] H. Boudali, *A Temporal Bayesian Network Reliability Modeling and Analysis Framework*, Ph.D. Dissertation, University of Virginia, (2005).

[Bou06] H. Boudali, "A Continuous-Time Bayesian Network Reliability Modeling and Analysis Framework," *IEEE Transactions on Reliability*, **55**, 86 (2006).

[Bre85] L. Breiman and J. H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, **80**, 580 (1985).

[Buc08] P. Bucci, J. Kirschenbaum, L. A. Mangan, T. Aldemir, C. Smith, and T. Wood, "Construction of Event Tree/Fault Tree Models form a Markov Approach to Dynamic System Reliability," *Reliability Engineering and System Safety*, **93**, 1616 (2008).

[Cat10] U. Catalyurek, B. Rutt, K. Metzroth, A. Hakobyan, T. Aldemir, R. Denning, S. Dunagen, and D. Kunsman, "Development of a Code-Agnostic Computational Infrastructure for the Dynamic Generation of Accident Progression Event Trees," *Reliability Engineering and System Safety*, **95**, 278 (2010).

[Cac08] D. G. Cacuci and M. Ionescu-Bujor, "Adjoint Sensitivity Analysis of Dynamic Reliability Models Based on Markov Chains – I: Theory," *Nuclear Science and Engineering*, **158**, 97 (2008).

[Che93] R. Chen and R. S. Tsay, "Nonlinear Additive ARX Models," *Journal of the American Statistical Association*, **88**, 955 (1993).

[Coj93] G. Cojazzi, P. C. Cacciabue, and P. Parisi, "DYLAM-3, A Dynamic Methodology for Reliability Analysis and Consequences Evaluation in Industrial Plants, Theory and How to Use," Euratom Report, EUR 15265 EN. (1993).

[Com82] "Zion Station Updated Final Safety Analysis Report," Commonwealth Edison Company (1982).

[Dem82] N. S. DeMuth, D. Dobranich, R. J. Henninger, *Loss of Feedwater Transients for the Zion-1 Pressurized Water Reactor*, Los Alamos National Laboratory Technical Report, LA-9296-MS (1982).

[Dev92] J. Devooght and C. Smidts, "Probabilistic Reactor Dynamics – I: The Theory of Continuous Event Trees," *Nuclear Science and Engineering*, **111**, 229 (1992).

[Dis09] S. Distefano and A. Puliafito, "Reliability and Availability Analysis of Dependent Dynamic Systems with DRBDs," *Reliability Engineering and System Safety*, **94**, 1381 (2009).

[Dog09] O. Doguc, J. E. Ramirez-Marquez, "A Generic Method for Estimating System Reliability using Bayesian Networks," *Reliability Engineering and System Safety*, **94**, 542 (2009).

[Don10] R. Donat, P. Leray, L. Bouillaut, and P. Aknin, "A Dynamic Bayesian Network to Represent Discrete Duration Models," *Neurocomputing*, **73**, 570 (2010).

[Dug93] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Fault Trees and Markov Models for Reliability Analysis of Fault-Tolerant Digital Systems," *Reliability Engineering and System Safety*, **39**, 291 (1993).

[Fri82] J.H. Friedman and W. Stuetzle, "Smoothing of Scatterplots," Technical Report ORION006, Stanford University (1982).

[Fus76] J.B. Fussell, E.F. Aber and R.G. Rahl, "On the Quantitative Analysis of Priority-AND Failure Logic," *IEEE Transactions on Reliability*, **R-25**, 324 (1976).

[Gal00] S. F. Galan and F. J. Diez, "Modelling Dynamic Causal Interactions with Bayesian Networks: Temporal Noisy Gates," *Proceedings of the 2nd International Workshop on Causal Networks*, Berlin, Germany (2000).

[Hab08] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek, "Dynamic Generation of Accident Progression Event Trees," *Nuclear Engineering and Design*, **238**, 3457 (2008).

[Hay99] M. R. Hayns, "The Evolution of Probabilistic Risk Assessment in the Nuclear Industry," *Transactions of the Institution of Chemical Engineers*, **77**, 117 (1999).

[Hof10] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, and M. Sonnenkalb, "Dynamic Event Trees for Probabilistic Safety Analysis," EUROSAFE Forum, Cologne, Germany (2010).

[Hol99] J. Holmberg, K. Hukki, L. Norros, U. Pulkkinen, and P. Pyyy, "An Integrated Approach to Human Reliability Analysis – Decision Analytic Dynamic Reliability Model," *Reliability Engineering and System Safety*, **65**, 239 (1999).

[Kel07] D. L. Kelly, "Bayesian Modeling of Time Trends in Component Reliability Data Via Markov Chain Monte Carlo Simulation," Idaho National Laboratory Technical Report, INL/CON-06-11953 (2007).

[Kel09] D. L. Kelly and C. L. Smith, "Bayesian Inference in Probabilistic Risk Assessment – The Current State of the Art," *Reliability Engineering and System Safety*, **94**, 628 (2009).

[Kim97] H. G. Kim and J. C. Lee, "Development of a Generalized Critical Heat Flux Correlation Through the Alternating Conditional Expectation Algorithm," *Nuclear Science and Engineering*, **127**, 300 (1997).

[Kol09] D. Koller and N. Friedman, *Probabilistic Graphical Models – Principles and Techniques*, MIT Press (2009).

[Kop05] V. Kopustinskas, J. Augutis, and S. Rimkevicius, "Dynamic Reliability and Risk Assessment of the Accident Localization System of the Ignalina NPP RBMK-1500 Reactor," *Reliability Engineering and System Safety*, **87**, 77 (2005).

[Kun08] D. M. Kunsman, S. Dunagan, T. Aldemir, R. Denning, A. Hakobyan, K. Metzroth, U. Catalyurek, and B. Rutt, *Development and Application of the Dynamic System Doctor to Nuclear Reactor Probabilistic Risk Assessments*, Sandia National Laboratories Technical Report SAND2008-4746 (2008).

[Lab00] P.E. Labeau, C. Smidts, and S. Swaminathan, "Dynamic Reliability: Towards an Integrated Platform for Probabilistic Risk Assessment," *Reliability Engineering and System Safety*, **68**, 219 (2000).

[Lab05a] P. E. Labeau and J.M. Izquierdo, "Modeling PSA Problems – I: The Stimuls-Driven Theory of Probabilistic Dynamics," *Nuclear Science and Engineering,* **150**, 115 (2005).

[Lab05b] P. E. Labeau and J. M. Izquierdo, "Modeling PSA Problems – II: A Cell-to-Cell Transport Theory Approach," *Nuclear Science and Engineering*, **150**, 140 (2005).

[Lan07] H. Langseth and L. Portinale, "Bayesian Networks in Reliability," *Reliability Engineering and System Safety*, **92**, 92 (2007).

[Lee06] C. Lee and K. J. Lee, "Application of Bayesian Network to the Probabilistic Risk Assessment of Nuclear Waste Disposal," *Reliability Engineering and System Safety*, **91**, 515 (2006).

[Lee11] J. C. Lee and N. J. McCormick, *Risk and Safety Analysis of Nuclear Systems*, John Wiley & Sons, Ltd. (2011).

[Mar96] M. Marseguerra and E. Zio, "Monte Carlo Approach to PSA for Dynamic Process Systems," *Reliability Engineering and System Safety*, **52**, 227 (1996).

[Mar10] D. Marquez, M. Neil, and N. Fenton, "Improved Reliability Modeling using Bayesian Networks and Dynamic Discretization," *Reliability Engineering and System Safety*, **95**, 412 (2010).

[Mon08] S. Montani, L. Portinale, A. Bobbio, and D. Codetta-Raiteri, "RADYBAN : A Tool for Reliability Analysis of Dynamic Fault Trees Through Conversion into Dynamic Bayesian Networks," *Reliability Engineering and System Safety*, **93**, 922 (2008).

[Mur02] K. P. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Dissertation, University of California, Berkeley (2002).

[Nea93] R. M. Neal, *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, Technical Report CRG-TR-93-1, University of Toronto (1993).

[NRC74] "Acceptance Criteria for Emergency Core Cooling Systems for Light-Water Nuclear Power Reactors," Title 10 Code of Federal Regulations Part 50.46, U. S. Nuclear Regulatory Commission (1974).

[NRC75] "Reactor Safety Study – An Assessment of Accident Risks I U. S. Commercial Nuclear Power Plants," WASH-1400, U. S. Nuclear Regulatory Commission (1975).

[NRC90a] "Overview and Comparison of U. S. Commercial Nuclear Power Plants," NUREG/CR-5640, U. S. Nuclear Regulatory Commission (1990).

[NRC90b] "Severe Accident Risks: An Assessment for Five U. S. Nuclear Power Plants," NUREG- 1150, U. S. Nuclear Regulatory Commission (1989).

[NRC07] "Industry-Average Performance for Components and Initiating Events at U. S. Commercial Nuclear Power Plants," NUREG/CR-6928, U. S. Nuclear Regulatory Commission (2007).

[NRC08] "Systems Analysis Program for Hands-On Integrated Reliability Evaluation (SAPHIRE), Technical Reference," NUREG/CR-6952, Vol. 2, U. S. Nuclear Regulatory Commission (2008).

[NRC01] "RELAP5/MOD3.3 Code Manual, Volume 1: Code Structure, Systems Models, and Solution Methods," NUREG/CR-5535, U. S. Nuclear Regulatory Commission (2001).

[Poc02] R. Pochard, F. Jedrzejewski, X. Mazauric, and P.Y. Carteron, "Analysis of a Feed and Bleed Procedure Sensitivity Study, Performed with the SIPACT Simulator, on a French 900 MWe NPP," *Nuclear Engineering and Design*, **215**, 1 (2002).

[Por10] L. Portinale, D. C. Raiteri, and S. Montani, "Supporting Reliability Engineers in Exploiting the Power of Dynamic Bayesian Networks," *International Journal of Approximate Reasoning*, **51**, 179 (2010).

[Pou08] O. Pourret, P. Naiim, and B. Marcot, *Bayesian Networks : A Practical Guide to Applications*, John Wiley & Sons, Ltd. (2008).

[Ran89] V. Ransom, *Course A – Numerical Modeling of Two-Phase Flow*, Course presented at Ecole d'ete d'Analyse Numerique, EGG-EAST-8546 (1989).

[She11] R. Sherry and J. Gabor, "Risk Informed Safety Margin Characterization: Trial Application to a Loss of Feedwater Event," *ANS PSA 2011 International Topical Meeting on Probabilistic Safety Assessment and Analysis*, Wilmington, NC (2011).

[Shi08] S. K. Shin and P. H. Seong, "Review of Various Dynamic Modeling Methods and Development of an Intuitive Modeling Method for Dynamic Systems," *Nuclear Engineering and Technology,* **40**, 375 (2008).

[Tor98] J. G. Torres-Toledano and L. E. Sucar, "Bayesian Networks for Reliability Analysis of Complex Systems," *Proceedings of the 6th Ibero-American Conference on AI: Progress in Artificial Intelligence* (1998).

[Wan04a] P. Wang and T. Aldemir, "Some Improvements in State/Parameter Estimation Using the Cell-to-Cell Mapping Technique," *Nuclear Science and Engineering*, **147**, 1 (2004).

[Wan04b] D. Wand and M. Murphy, "Estimating Optimal Transformations for Multiple Regression Using the ACE Algorithm," *Journal of Data Science*, **2**, 329 (2004).

[Web06] P. Weber and L. Jouffe, "Complex System Reliability Modelling with Dynamic Object Oriented Bayesian Networks (DOOBN)," *Reliability Engineering and System Safety*, **91**, 149 (2006).

[Web10] P. Weber, G. Medina-Oliva, C. Simon, and B. Lung, "Overview on Bayesian Networks Applications for Dependability, Risk Analysis and Maintenance Areas," *Engineering Applications of Artificial Intelligence*, Article in Press (2010).

[Wes93] "Emergency Response Guidelines," Westinghouse Electric Corporation(1993).

[Xu09] H. Xu, L, Xing, and R. Robidoux, "DRBD: Dynamic Reliability Block Diagrams for System Reliability Modelling," *International Journal of Computers and Applications*, **31**, 132 (2009).

[Yug08] T. Yuge and S. Yanagi, "Quantitative analysis of a Fault Tree with Priority AND Gates," *Reliability Engineering and System Safety*, **93**, 1577 (2008).

[Zag99] A. T. Zagorecki, *Local Probability Distributions in Bayesian Networks: Knowledge Elicitation and Inference*, Ph.D. Dissertation, Bialystok University of Technology (1999).

[Zio11] E. Zio and N. Pedroni, "How to Effectively Compute the Reliability of a Thermal-Hydraulic Nuclear Passive System," *Nuclear Engineering and Design*, **241**, 310 (2011).