UNCERTAINTY QUANTIFICATION FOR REACTOR SAFETY ANALYSIS

by

Douglas A. Fynan

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
2014

Doctoral Committee:

Professor John C. Lee, Chair
Professor Kenneth Powell
Professor William R. Martin
Associate Professor Annalisa Manera
Kwang-Il Ahn, Korea Atomic Energy Research Institute

To Michelle

ACKNOWLEDGEMENTS

I would like to thank my doctoral committee and the faculty and staff in the Department of Nuclear Engineering and Radiological Science at the University of Michigan for their guidance and support over my graduate school career. Much of the work in this dissertation was supported by the Korea Atomic Energy Research Institute and I would like to especially thank Kwang-Il Ahn and Joon-Eon Yang of the Integrated Safety Assessment Division for making a summer research opportunity for me at KAERI possible. I would also like to thank Tae Woon Kim and many other members at KAERI for a wonderful professional and cultural experience.

I would like to recognize two scholarships, the Class of 1931E scholarship and the Rogel scholarship, for providing tremendous financial support to me making my undergraduate education at the University of Michigan possible. Special thanks to John Hallenburg, Kate Weber, Jack Schultz, and Bill Berglin, all of whom have uniquely contributed to my life over the years. Thank you to Grandma and Grandpa, Uncle Mark, and Uncle Steve for your continued love and support.

Finally, I would like to thank Professor John C. Lee for his many years of guidance and commitment to me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 System TH Codes for Reactor Safety Analysis

Safety analysis is performed for nuclear power plants (NPPs) to assess the effectiveness of safety systems in NPP design and to calculate the consequences of reactor accidents. A tool essential to safety analysis are system thermal-hydraulic (TH) computer codes used to model the transient behavior of the NPP and safety systems during accidents. Over the past 40 years, TH codes have evolved from conservative evaluation models used to analyze Design Basis Accidents (DBAs) for NPP licensing to best-estimate (BE) TH codes such as RELAP5, MARS, TRACE, CATHERE, and ATHLET that can realistically model complex physical processes and phenomena that can occur in a NPP. TH codes have broad applications in the nuclear industry including NPP licensing, Probabilistic Safety Assessment (PSA), severe accident research, design and analysis of experiments, and establishing Emergency Operating Procedures (EOPs) and Severe Accident Management Guidelines (SAMGs) [Aur12].

Historically, TH codes have been computationally expensive requiring state-of-the-art computer resources, extensive storage, and long computation times. The computational requirements for TH calculations were a constraint. Over the past two decades, advancements in computer science and technology including the personal computer, exponentially faster processors, multi-core processors, and parallel computing have made performing tens to hundreds of detailed transient calculations with TH codes tractable at the desktop level for all engineers. Large parallel to massively parallel computing and supercomputers make the ability to perform tens of thousands calculations a near term reality. While brute force application of

ever expanding computational power to reactor safety problems may be enticing, efficient use of resources is always the prudent route since engineering judgment is required to interpret simulation results and draw meaningful conclusions. The computing power available today and tomorrow gives an extra degree of freedom to engineers in approaching reactor safety problems enabling new or improved methods for licensing calculations and PSA.

Along with the increased availability of computational resources comes increased demand for TH codes and new applications that challenge the capabilities of the codes. In the United States, the ageing NPP fleet is operated at higher power from power uprates, to higher fuel burnups, and on longer fuel cycles. Changes to the safety margin of NPPs must be rigorously assessed considering ageing and operational issues. Figure 1.1 illustrates the concept of safety margins for NPPs that TH codes are regularly used to calculate. Furthermore, new TH problems such as spent fuel pool accidents are analyzed with TH codes originally developed to simulate loss-of-coolant accidents (LOCAs).

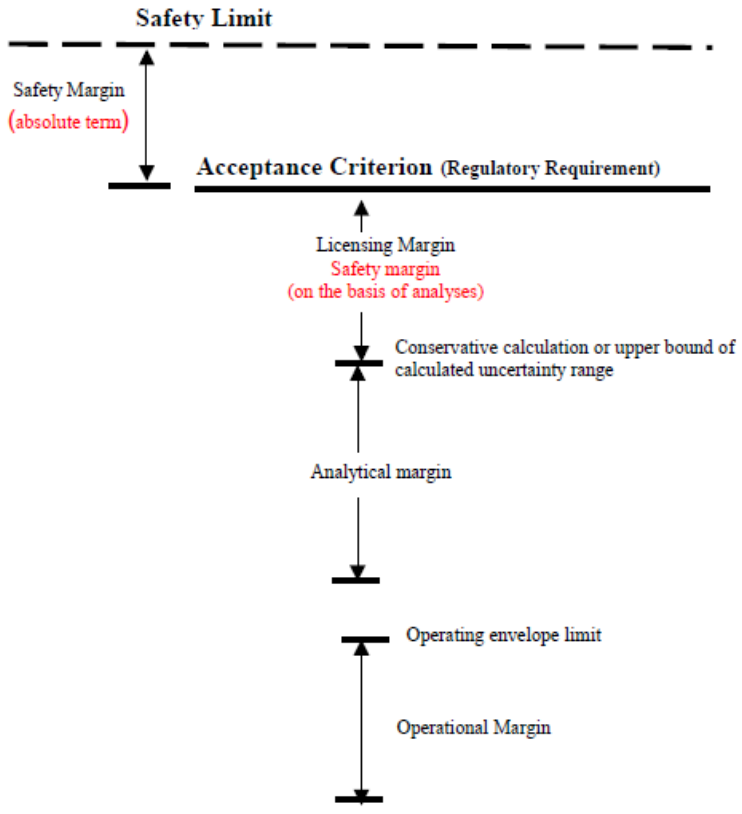

Figure 1.1. Safety margins for nuclear power plants. From [IAE04].

1.2 Uncertainty Quantification of Computer Experiments

Physical experiments are invaluable in science and engineering providing direct measurement of properties of natural or engineered systems. In many fields, experiments are expensive requiring the construction of experimental facilities, complex measurement devices, and prototypes and in some applications, can pose real safety risks to the experimenters and the environment, e.g., test pilots flying experimental aircraft. Cost and safety are particular constraints to reactor safety experiments where the use of radioactive material introduces unavoidable liabilities. Most experimental results are irreproducible; the same experiment when repeated will yield different results due to continuously changing environmental variables and measurement error.

With advancement of computer science and technology, computer codes and simulation have become the go to tool across science and engineering disciplines, supplementing and sometimes replacing experiment. Computer simulations or computer experiments are usually repeatable and give an "exact" numerical result up to machine precision. The fidelity of computer models and predictive accuracy have evolved to the point where simulation results may be more reliable than experiment in some situations. However, the aforementioned challenges with physical experiments, cost and safety, are often driving the transition so simulation based engineering analysis and design. In reactor safety, the loss of experimental facilities and thus the capacity to perform experiments from decommissioning for a variety of reasons (cost, political, environmental) forces the transition.

Computer codes attempt to accurately describe the underlying physics and behavior of natural or engineered systems through mathematical models. These models can be derived from first principles, theory, semi-empirical, empirical derived, or a combination of these. The physics of the problem are always simplified or approximated by the code models. The system being modeled is also simplified or approximated by assumptions made in the code and the solution techniques. For example, three-dimensional flow may be approximated to one-dimension and truncation errors are introduced in finite differencing schemes through nodalization and time step size. Furthermore, external data supplied to the code models, cross sections, fission product yields, material properties, heat transfer coefficients, etc., are not known values and have been experimentally measured (with error) or estimated from another model.

The assumptions in computer codes and models makes every computer simulation an *estimate* of the system being modeled. The estimate has bias, error, and *uncertainty*, all of which need to be quantified when using simulations in making an engineering decision. In the context of this dissertation, bias is defined as a systematic under or over prediction of a model or the deliberate neglect of a process or phenomena in a model. For example, assuming one-dimensional flow in a reactor core is a bias. Error is introduced from the level of refinement chosen for the system description as compared to an asymptotic solution obtained at the maximum refinement. Nodalization, time step size, convergence criteria, and number of histories in Monte Carlo calculations are sources of error. The bias and error of a simulation are generally functions of the models and solution techniques chosen for the problem and the acceptable magnitude of bias and error is problem specific. Some applications may require a higher fidelity model at higher computational cost or a physical experiment. For some reactor safety problems, bias may be acceptable or even intentionally introduced as long as the bias is conservative with respect to the calculation of the safety parameter of interest.

The uncertainty of a computer experiment is a measure of the precision of the models and the model input parameters used in the simulation. Input parameters are both the external data supplied to the models and parameters defining the operating envelope of the system being analyzed. The uncertainty of the input parameters can be described by probability density functions (PDFs) defining the range and associated probabilities of values that an input parameter can take. Uncertainty quantification (UQ) asks the question: what range of outputs will be observed given the range of uncertain input parameter values? UQ is the formal process of determining the range and probabilities of the outputs or the output PDF. UQ allows engineering decisions to be made using the results of computer experiments while acknowledging the result is an estimate. UQ can be used to answer the questions: how good is the estimate and is the estimate good enough for the application?

Figure 1.2 illustrates the UQ process for a computer experiment. UQ involves propagating the uncertainty of the input parameters, described by their respective PDFs, through the computer code to obtain the output parameter PDF. A computer code can be interpreted as a very large, complex nonlinear function of the input parameters. Analytical solutions only exist for propagating PDFs through very simple functions so UQ methodologies must be utilized to either approximate the output PDF or obtain estimates of the statistical properties, the mean,

variance, kurtosis, 95th percentile, etc., of the PDF. Three classes of UQ methodologies are sampling based methods, code surrogates, and adjoint methods. Adjoint methods require looking into where and how the input parameters appear in the mathematical models embedded within the code and the how the numerical methods solve the equations in order to calculate derivative and gradient information. The models and the numerical methods used can be very complicated and implemented in thousands of lines of computer code so adjoint methods are difficult to implement especially at an engineering design level. Sampling based methods and code surrogates use the computer code in black box mode using only the input and output streams from the simulation.

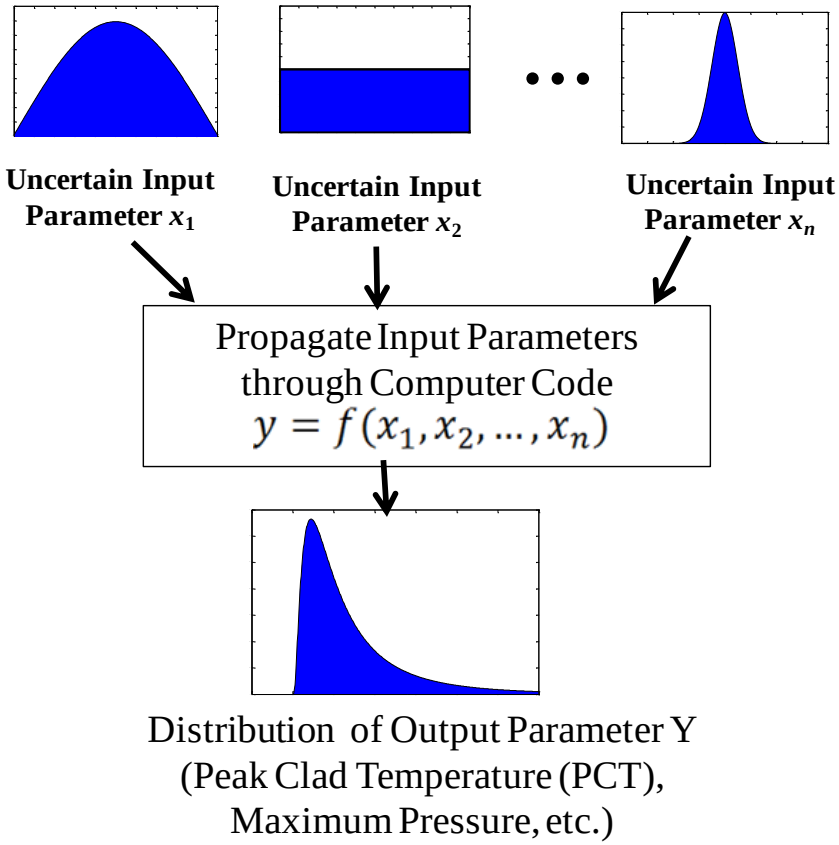

Figure 1.2. Illustration of uncertainty quantification for computer experiments.

Sampling based methods sample the input parameter distributions and execute the code at the sampled input values. The simplest sampling strategy is the Monte Carlo (MC) method where the input PDFs are randomly sampled many times and the output PDF is constructed by

tallying the output of each simulation into a distribution. For multi-dimensional input parameter spaces characteristic of many engineering problems including NPP transients, tens of thousands to millions of calculations would be required to obtain a statistically significant result from direct MC simulation making the MC method computationally impractical for UQ with TH codes. Statistical information about the output PDF must be estimated from samples of limited size. For example, nonparametric order statistics, namely Wilks' formula [Wil41], use random sampling to obtain confidence limits of the output PDF with limited sample size.

An alternative to the MC method is Latin hypercube sampling (LHS) [McK79]. LHS is a variance reduction technique for the MC sampling. LHS designs subdivide each dimension in the input parameter space into *N* equal probability intervals or strata. Data points are generated by randomly sampling one parameter value from within each interval and randomly combining with values from the other dimensions. The resulting *N* data points ensure all portions of the input PDFs are sampled. Figure 1.3 illustrates a four data point LHS design with two input parameters. The mean and variance of the output PDF can be estimated from the sample of output values.

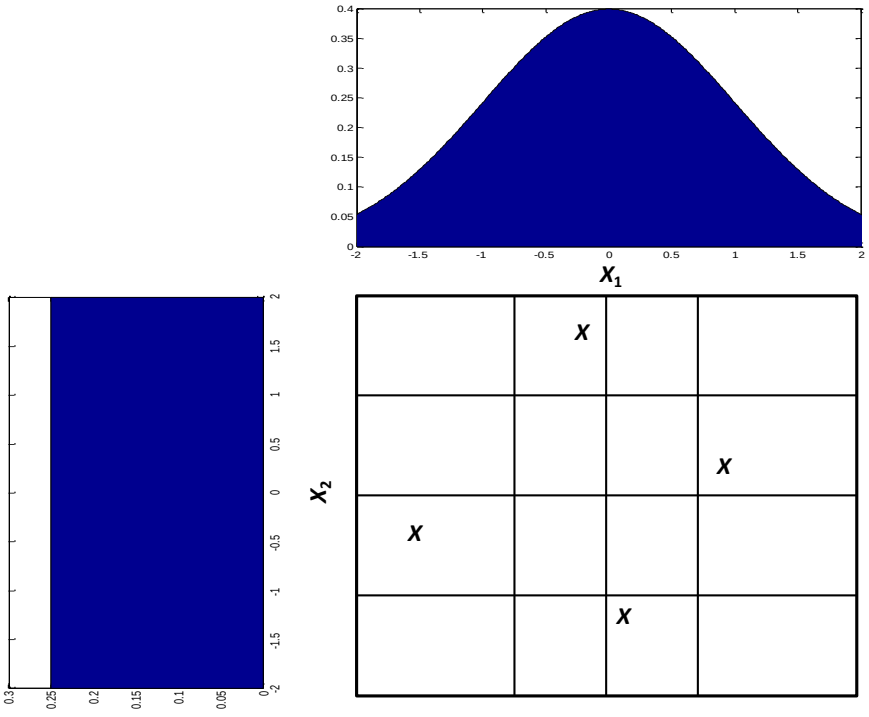

Figure 1.3. LHS design with two input dimensions and four strata.

The last UQ method, code surrogates, are a special case of sampling. Code surrogates are simplified mathematical models of the functional relationship between the inputs and outputs. Code surrogates are constructed through regression analysis on a limited number of code simulations, a data set of input parameter values and associated output response. Once the surrogate is constructed, it is a fast-running approximation to the computer code and can be exhaustively sampled through a MC method to obtain an approximate output PDF at greatly reduced computational cost. However, surrogate construction is a nontrivial task and the accuracy of the surrogate must be rigorously assessed and quantified. The uncertainty of the surrogate, itself an UQ tool, must be quantified.

1.3 Research Scope and Objectives

In this dissertation, our main goal is to develop and demonstrate new techniques based on code surrogates and *deterministic* sampling strategies for UQ in reactor safety analysis. Specifically, the techniques will address some of the computational challenges that limit the application of dynamic PSA methodologies to realistic NPP transients and simulation with system TH codes. Our objectives are:

- To demonstrate a new methodology to develop a dynamic code surrogate that can accurately simulate time dependent, nonlinear TH behavior of a NPP transient considering multiple safety system degradations or failures. The methodology applies an existing nonparametric regression technique, the Alternating Conditional Expectation (ACE) algorithm, in a machine learning application to construct a discrete time dynamic system model that can replace a system TH code in a dynamic PSA model.
- Quantify the uncertainty of the dynamic code surrogate. A derivation of the variance of the ACE algorithm transformations provides a consistent estimation of the surrogate model uncertainty. Using the Unscented Transform (UT), an existing deterministic algorithm for uncertainty propagation and system state estimation in nonlinear dynamic systems, the time dependent uncertainty of the system state trajectory predicted by the dynamic code surrogate is calculated.
- Demonstrate the accuracy of the UT as general UQ methodology for problems with multi-dimensional input parameter spaces. The UT is a deterministic sampling based

strategy that scales linearly with the size of the input parameter space and can provide accurate estimates of the mean and variance of the output parameter PDF.

- Apply the dynamic code surrogate and UT to a dynamic PSA study of a realistic NPP transient. The recirculation phase of the hot leg large break LOCA (HL-LBLOCA) in a pressurized water reactor (PWR) is studied. The subcooled water level in the core during the HL-LBLOCA is predicted with the dynamic code surrogate. Multiple degradations in the high-pressure safety injection (HPSI) system and the containment sump are sampled with the UT and benchmarked against MC simulation with the surrogate.

In Chapter 2, an overview of dynamic PSA for NPPs will be given. The current limitations of use of system TH codes in dynamic PSA are discussed. Chapter 3 will discuss engineering applications of code surrogates and the UT. An overview of regression analysis will presented followed by detailed derivations of the Gaussian Process Model (GPM), ACE algorithm, and the UT. All three methods are then applied to the UQ of the peak clad temperature (PCT) during a LBLOCA, a classical example of UQ in Best Estimate Plus Uncertainty (BEPU) methodologies used for NPP licensing. Chapter 4 describes the Ulchin 3&4 NPP, RELAP5 system TH code, and the LBLOCA RELAP5 model of Ulchin 3&4 which is used throughout this study. Chapter 5 presents the methodology for the dynamic code surrogate construction including input parameter selection and benchmarking against RELAP5 results for the recirculation phase of the HL-LBLOCA. Chapter 6 is the dynamic PSA study of the HL-LBLOCA implementing the dynamic code surrogate and the UT to assess safety system degradations.

References

[Aur12] F. D'Auria, "Perspectives in SystemThermal-Hydraulics," *Nuclear Engineering & Technology*, **44**, 855 (2012).

[IAE04] IAEA, Implications of power uprates on safety margins of nuclear power plants, IAEA-TECHDOC-1418, IAEA, Vienna, Austria, 2004.

[McK79] M.D. McKay, R.J. Beckman and W.J. Conover, "Comparison of Three Methods for Selecting of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **21**, 23(1979).

[Wil41] S. S. Wilks, “Determination of Sample Sizes for Setting Tolerance Limits,” *The Annals of Mathematical Statistics*, **12**, 91 (1941).

CHAPTER II

DYNAMIC PROBABILISTIC SAFETY ASSESSMENT

Probabilistic safety assessments are systematic studies that calculate probabilities and consequences of accidents that could occur at NPPs. A PSA estimates the risk, the probability times the consequence of an event, for the NPP being studied and provides insight into the strengths and weaknesses of the NPP design and operation procedures. In the framework of PSA, the static event tree (ET) and fault tree (FT) approach is used to model the accident sequences from an initiating event to the end state. TH codes are used to calculate the physical response of the NPP and determine the end state to the order of events and equipment states preset by the analyst in the ET/FT approach. For decades, ET/FT methodologies have been a useful tool in PSA applications; however, limitations of the ET/FT to model coupling between failure events, operator actions, and the dynamic system states have been recognized [Sui94]. The accident at Three Mile Island Unit 2 (TMI-2) clearly demonstrated the need to consider dynamic dependencies in ET/FT methodologies [Sui94]. Consequently, dynamic PSA methodologies have been under development to capture stochastic behavior arising from the interaction, coupling, and dependencies of plant dynamics with human operators, control systems and degradation of safety system function. Dynamic PSA methodologies attempt to couple the probabilistic nature of failure events, component transitions, and human reliability to deterministic calculations of the time-dependent plant response. Due to the overall complexity of PSA and NPP systems, no single method is sufficient for all situations so dynamic methodologies supplement the conventional ET/FT approach in PSA.

2.1 Overview of Dynamic PSA Methodologies

Dynamic PSA methodologies can be divided into two broad groups: continuous-time methods and discrete-time methods. The first continuous-time method introduced for reactor

dynamics problems was the continuous event tree (CET) [Dev92]. The CET is derived from a stochastic balance equation obtained from the differential Chapman-Kolmogorov equation describing the transition rate of the component state or system configuration as a function of time. The transitions are assumed to be governed by Markovian processes. The integral form of the balance equation derived in [Dev92, Lee11] gives a joint PDF $p(x,c,t)$ for the system and component states as a function of time

$$p(x,c,t) = \int dx' p(x',c,0)\delta[x - g(x',c,t)] \exp\left\{-\int_0^t ds\Gamma[g(x',c,s)]\right\} + \int_0^t dt' \int dx' \int dc' p(x',c',t')\delta[x - g(x',c,t-t')] \times \exp\left\{-\int_{t'}^t ds\Gamma[g(x',c,s-t')]\right\} W(c' \to c, x') \quad . \tag{2.1}$$

The probability per unit time of the discrete component state transition $c' \to c$ is W given system state x'. The total probability per unit time of leaving component state c is

$$\Gamma(c,x) = \int W(c \to c', x) dc' \quad . \tag{2.2}$$

The system state trajectory $x(t)$ is uniquely determined from a previous system state x' and component state $c(t)$ through the function $g()$ which represents a deterministic TH code simulation

$$x(t) = g(x',c,t) \quad . \tag{2.3}$$

Equation (2.1) representing the CET is mathematically complicated due to the deterministic calculation of Eq. (2.3) which is embedded within multiple integrals in Eq. (2.1) and the system state dependence of the component transition W. Only after problem specific simplifications can one begin to Eq. (2.1), usually requiring Monte Carlo methods. The continuous cell-to-cell-mapping technique (CCCMT) [Tom97] is a version of the CET that partitions the system state into discrete cells. Cell boundaries are usually defined by setpoint values of system state variables that would trigger component state changes such as pressure setpoints for safety relief valves. Due to the complexity and extensive computation requirements of the continuous-time

methods, applications of these methods to realistic full-scale nuclear systems modeled by system TH codes have been limited [Dev92a,Kop05,Ald13].

Discrete-time methods for dynamic PSA are direct Monte Carlo simulation, dynamic event tree (DET), and Markov/cell-to-cell-mapping technique (Markov/CCMT). The later modifies the CCCMT to a discrete-time formulation by assuming all component states will remain constant over an user defined time steps. Cells are discretized in both the system state variables and time domains. Direct Monte Carlo simulation samples component transitions at discrete time steps to produce branching in the system evolution. The summation of each modeled sequence and respective cumulative branch probability can give the frequency of end states.

The most common discrete-time method is the DET. In conventional ET analysis, the timing and sequence of branches representing system failures and successes are generally preset by the analyst and the system response for each branch is calculated. In a DET, branching rules and conditions are defined by the analyst. As the system evolves from the initiating event, branches are generated when a system state parameter crosses a setpoint or operator action is required. The DET realistically models the timing of the component, system, and operator action success or failure as demanded by the evolution of the system state for each sequence of branches.

The first DET methodology developed was the Dynamic Logical Analytical Methodology (DYLAM) [Coj96]. DYLAM allows both demand type and stochastic failures and transitions to be modeled with constant or functional dependent rates. A fixed time step Δt is assumed for the analysis and all failures and transitions can only occur at multiplies of Δt so branch points can only occur at discrete times. If a demand type setpoint is reached during a time step, a branch is generated at the next branch time. For stochastic transitions, transition probabilities for all components are generated at every branch time. If a transition probability exceeds an user defined threshold, a branch is generated. Many software programs implementing DYLAM-based methodologies such as Dynamic Event Tree Analysis Method (DETAM) [Aco93], Accident Dynamic Simulator (ADS) [Hsu96], Monte Carlo Dynamic Event Tree (MCDET) [Klo06], and Analysis of Dynamic Accident Progression Trees (ADAPT) [Hak08, Cat10] have been developed for DET analysis. While the implementation of specific branching rules can vary, most programs are similar in architecture coupling the generation of

branching to deterministic calculations of the time-dependent plant system state while tracking the probabilities of each path.

2.2 System Modeling in Dynamic PSA

A key computational requirement of all dynamic PSA methodologies is the calculation of the accident sequences with a time-dependent system model. For even a small problem, the system evolution for thousands of branches may need to be potentially evaluated. In DET programs, the system model is usually evaluated as a subroutine in the calculation scheme with the DET program directing new system model executions. As branch points are generated, the component states in the system model are updated and the new calculations must be performed following each new branch. System state information is then passed back from the system model to the DET program to calculate any additional branching at the next calculation time step.

For NPP problems, transient calculations are usually performed by system TH codes such as RELAP5 or MELCOR. Components such as valves and pumps can be represented by dedicated models or as boundary conditions to the system nodalization. Control functions are used to actuate or stop components and define operational properties. Some DET programs offer direct coupling to TH codes. MCDET is hardwired to MELCOR and ADAPT can be linked to both RELAP5 and MELCOR. These programs pause the calculation at each DET time step, extract necessary system state information from the simulation output and overwrites control function logic in the input deck if component transitions need to be modeled. Then new simulations following all new branches are restarted from the paused calculation.

Coupling to TH codes require extensive use of output and restart files that are generated during transient calculations. A single restart file can be gigabyte-sized so management of the data streams between the TH code calculations, which on their own are computationally expensive to obtain, and the DET program is a nontrivial task as the number of sequences being calculated grows with each additional branch of the DET. ADAPT attempts to manage the computational load with massively parallel processing now available with modern computing architectures [Cat10].

2.3 The Curse of Dimensionality

In the dynamic PSA methodologies discussed in previous sections, components are assumed to occupy discrete states. The simplest component state model is a binary system (nominal/failed, on/off, open/closed, etc.). Some components can occupy a continuous spectrum of states so a binary description may not be adequate. For example, a valve that is demanded to be fully open could be considered in any number of degraded states if the valve only partially opens between 0% and 100% of the nominal flow area. A high pressure safety injection system that automatically starts at full flow capacity can be manually throttled to a reduced flow by operator action as was the case in the TMI-2 accident. In accident sequences where partial system degradations could be important, a refined discretized component state description may be needed.

For a system comprised of multiple components, the total number of component state combinations N follows a power law formula given by

$$N = \prod_k k^{n_k} \tag{2.4}$$

where k is the number of discrete states and n_k is the number of components discretized with k states. If a system is only comprised of n binary components, Eq. (2.1) simplifies to 2^n unique combinations of component states. As a system becomes more complex with an increasing number of components modeled or refined component state discretization, the consequence of Eq. (2.4) is the explosion of component state combinations. In a DET, every component state combination at each time step can represent a branch so the number of possible sequences increase geometrically as the number of time steps increases. An early application of DETAM to a steam generator tube rupture (SGTR) problem assumed 7 binary systems for a total of 128 component states, 324 crew planning states, and 2304 crew diagnostic states for a total of 9.6 $\times 10^7$ distinct component states at each time step [Aco93]. The curse of dimensionality in both the component state space and time domain is a fundamental limitation of DET and other dynamic methodologies when applied to realistic NPP systems.

In order to limit the total number of accident sequences that need to be analyzed and manage the size of the DET, DET methodologies must apply simplifying assumptions to the failure modes and transitions. Failure modes for components can generally be classified as on

demand or in time failures. On demand failures only generate branching when a system state setpoints or operator action criteria are satisfied while in time failures are stochastic in nature and can occur at any time step after the successful start of a system. To limit the number of in time branching in the DET, branches are only generated after the time-dependent cumulative distribution function for the failure of the component in its current state exceeds a probabilistic threshold provided by the user. Simultaneous failure of multiple components are not considered. Lastly, repair transitions for some components are not allowed so once those components transition to a failed state, they remain failed for the rest of the time steps.

The second method to manage the size of the DET is to incorporate stopping rules that terminate particular sequences and prunes them from the DET. The first type of stopping rule is a probability cutoff that terminates a sequence once the cumulative branch probability falls below a minimum probability value. Probability cutoffs prevent computational resources from being spent on very low probability sequence. The second stopping rule is halting sequences if an absorbing state is reached. An absorbing state occurs when the system state reaches a particular high-level safe or unsafe state. For example, the SGTR problem used successful completion of reactor coolant system (RCS) cooldown and depressurization or successful depressurization through feed and bleed cooling as safe absorbing states and steam generator dryout or failure to initiate feed and bleed when demanded as unsafe absorbing states. The last stopping rule is grouping which allows two sequences which are in similar hardware and operator crew state to be grouped together at any time step if their system states such as primary pressure and temperature are approximately equal.

Successful application of simplifying assumptions and stopping rules can generate manageable DET from which meaningful conclusions about realistic NPP transients can be drawn as demonstrated by the SGTR problem that ultimately calculated only 52 scenarios out of a possible sequences numbering in the hundreds of millions [Aco93]. However, the accuracy of a dynamic PSA is a function of the number of sequences analyzed, the size of the time steps and refinement of component states that represent all realistic failure and degradation modes. Furthermore, low probability sequences can explore unique regions of the joint system, component, and human reliability state space and could have large consequences so they should not be completely eliminated from the analysis. The size of the DET must be balanced with the computational expense of running deterministic TH calculations. Any new methods that could

help alleviate the dimensional explosion of branches in state space and time domain or reduce the computational expense of TH calculations would help the application of dynamic PSA to NPP systems.

The objective of this dissertation is to demonstrate the feasibility of two methods, dynamic code surrogates and the unscented transform, and their applicability to reactor safety analysis. Both methods address current computational challenges of dynamic PSA, unavoidable deterministic TH calculations and how to obtain an accurate PSA result while sampling a large input parameter space (branch times and component states transitions) efficiently. Chapter 6 presents a dynamic code surrogate and the unscented transform in a DET framework for the recirculation phase of a LBLOCA.

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).

[Ald13] T. Aldemir, "A Survey of Dynamic Methodologies for Probabilistic Safety Assessment of Nuclear Power Plant," *Annals of Nuclear Energy*, **52**, 113 (2013).

[Cat10] U. Catalyurek, B. Rutt, K. Metzroth, A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, 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).

[Coj96] G. Cojazzi, "The DYLAM Approach for the Dynamic Reliability Analysis of Systems," *Reliability Engineering and System Safety*, **52**, 279 (1996).

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

[Dev92a] J. Devooght and C. Smidts, "Probabilistic Reactor Dynamics II: A Monte Carlo Study of a Fast Reactor Transient," *Nuclear Science and Engineering*, **111**, 241 (1992).

[Hak08] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U.

Catalyurek, "Dynamic Generation of Accident Progression Event Trees," *Nuclear Science and Engineering*, **238**, 3457 (2008).

[Hsu96] K.S. Hsueh and A. Mosleh, "The Development and Application of the Accident Dynamic Simulator for Dynamic Probabilistic Risk Assessment of Nuclear Power Plants," *Reliability Engineering and System Safety*, **52**, 297 (1996).

[Jcl11] J.C. Lee and N.J. McCormick, *Risk and Safety Analysis of Nuclear Systems*, John Wiley & Sons, Inc., Hoboken, New Jersey (2011).

[Klo06] M. Kloos and J. Peschke, "MCDET: A Probabilistic Dynamics Method Combining Monte Carlo Simulation with the Discrete Dynamic Event Tree Approach," *Nuclear Science and Engineering*, **153**, 137 (2006).

[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).

[Sui94] N. Sui, "Risk Assessment for Dynamic Systems: An Overview," *Reliability Engineering and System Safety*, **43**, 43 (1994).

[Tom97] B. Tombuyses and T. Aldemir, "Computational Efficiency of the Continuous Cell-to-Cell Mapping Technique as a Function of Integration Schemes," *Reliability Engineering and System Safety*, **58**, 215 (1997).

CHAPTER III

CODE SURROGATES FOR ENGINEERING APPLICATIONS

In the previous chapters, the use of system TH codes for NPP accident analysis was presented. Decades of active research in nuclear safety, TH code development, and computer science has produced a suite of TH codes that is readily available to engineers for a variety applications in the nuclear industry. However, the complexity of nuclear systems and the need to meet rigorous safety regulations poses nontrivial challenges to NPP accident analysis. Chapter 2 discussed the computational challenges associated direct use of system TH codes in dynamic PSA.

In this chapter, we introduce code surrogates as fast running models to computationally expensive TH codes. First, an overview of code surrogates for engineering applications is given. Next, two nonparametric surrogate models, the Gaussian Process Model (GPM) and Alternating Conditional Expectation (ACE) algorithm are derived. The Unscented Transform (UT) is presented as an alternative sampling based tool for uncertainty quantification. Finally the historical use of code surrogates in NPP licensing is examined and contemporary applications of the GPM, ACE algorithm, and UT to a LBLOCA is presented.

3.1 Overview of Code Surrogates

Code surrogates are mathematical models that approximate the input/output relationship of a more complex computer code simulation. Surrogates can also be used to approximate the outcome of a physical experiment but the discussion will be limited to computer experiments. Alternative expressions for code surrogates are response surface models, metamodels, and code emulators. Code surrogates are fast running and are computationally inexpensive to execute making them attractive to use in applications such as design optimization and uncertainty quantification where many simulations need to be performed and direct use of the computer code would be computationally prohibitive. Many code surrogate models and algorithms are readily

available in open source and commercial scientific computing packages such as R, Matlab, and the DAKOTA project [Ada09].

Surrogate models must be constructed through regression analysis. Regression analysis estimates the input/output relationship from a data set obtained from a finite number of code simulations. Surrogate models can be grouped into two broad classes of regression analysis, parametric and nonparametric methods. Parametric regression assumes the functional form of the input/output relationship and specific function parameters and coefficients are learned through regression of the data. Linear regression, nonlinear regression, and spline models are examples of parametric regression. Nonparametric techniques make no assumptions about the functional form of the input/output relationship and the regression model is learned from properties of the data set. A broad spectrum of models including multivariate adaptive regression splines (MARS) [Fri91], artificial neural networks (ANNs), Gaussian process models (GPMs) and alternating conditional expectation (ACE) algorithm [Bre85] are nonparametric. Parametric regression is well suited for problems where some prior information about the underlying function form is known while nonparametric regression works best on large data sets from which inferences can be made.

3.1.1 Linear Regression Models

The goal of regression analysis is to estimate the relationship $y = f(\boldsymbol{x}) + \varepsilon$ where $\boldsymbol{x} = \{x_1, x_2, \ldots, x_p\}^T$ and ε is model noise given a data set $(X,\boldsymbol{y}) = \{(x_{11},..,x_{1p}, y_1);\ldots;(x_{n1},..,x_{np}, y_n)\}$. A linear regression model assumes the relationship is linear in fitting coefficients β_i:

$$f(\boldsymbol{x}) = \beta_0 + \beta_1\phi_1(\boldsymbol{x}) + \beta_2\phi_2(\boldsymbol{x}) + \cdots + \beta_m\phi_m(\boldsymbol{x}) \ . \tag{3.1}$$

Each ϕ_i is an assumed basis function of $\boldsymbol{x}$. For each data point $(\boldsymbol{x}_j, y_j)$, Eq. (3.1) is

$$y_j = \beta_o + \beta_1\phi_1(\boldsymbol{x}_j) + \beta_2\phi_2(\boldsymbol{x}_j) + \cdots + \beta_m\phi_m(\boldsymbol{x}_j) + \varepsilon_j \ . \tag{3.2}$$

Expressed in matrix notation,

$$\boldsymbol{y} = \Phi^{\mathrm{T}}\beta + \varepsilon \qquad \text{with} \tag{3.3}$$

$$\boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \Phi^{\mathrm{T}} = \begin{pmatrix} 1 & \phi_1(\boldsymbol{x}_1) & \cdots & \phi_m(\boldsymbol{x}_1) \\ 1 & \vdots & \ddots & \vdots \\ 1 & \phi_1(\boldsymbol{x}_n) & \cdots & \phi_m(\boldsymbol{x}_n) \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_m \end{pmatrix}, \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

The optimal fitting coefficients $\hat{\beta}$ that minimize the error term ε can be calculated through ordinary least squares. The resulting equations for $\hat{\beta}$ and the fitted regression model $\hat{f}(\boldsymbol{x})$ that approximates the true function $f(\boldsymbol{x})$ are

$$\hat{\beta} = \left(\Phi\Phi^{T}\right)^{-1}\Phi y \quad \text{and} \tag{3.4}$$

$$\hat{f}(\boldsymbol{x}) = \hat{\beta}_0 + \hat{\beta}_1\phi_1(\boldsymbol{x}) + \hat{\beta}_2\phi_2(\boldsymbol{x}) + \cdots + \hat{\beta}_m\phi_m(\boldsymbol{x}) \ . \tag{3.5}$$

The straight forward matrix inversion and subsequent matrix operations given in Eq. (3.4) and flexibility to use any number and types of basis functions ϕ result in the common use of linear regression models as code surrogates.

A specialized class of linear regression models are spline models. Spline models are piecewise functions usually comprised of low-degree polynomial functions called splines. The input parameter space is partitioned into subspaces and a spline is fitted to the subset of data points residing in each subspace. Continuity requirements for the function value and derivates can be applied at the knots where splines connect at subspace boundaries. Spline models allow relatively complex nonlinear functions to be approximated with simple low-degree polynomial functions while avoiding numerical instabilities that are often encountered with fitting a single high-degree polynomial function to an entire input space. Figure 3.1 shows an example of a spline model with one knot, continuity requirement for only the function value, and using polynomials of degree one to approximate a piecewise linear function with additive noise ε distributed as $N(0,0.25)$. The fitted spline model closely approximates the underlying function. MARS is a nonparametric extension of spline models. The number of splines and location of knots are learned from data through a recursive partitioning method [Fri91].

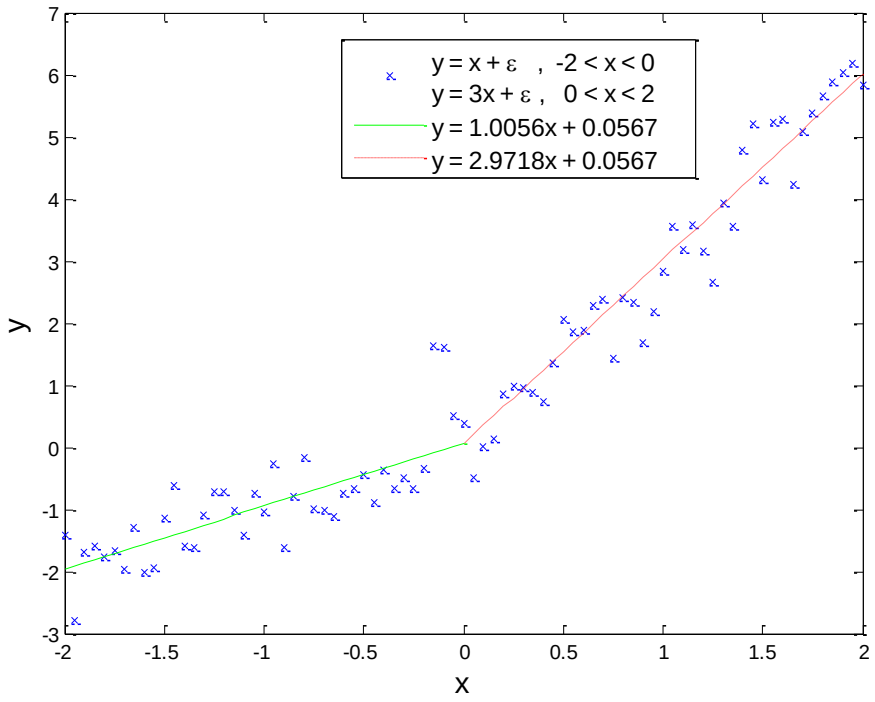

Figure 3.1. Spline model with one knot at $x = 0$ fitted to noisy data sampled from a piecewise linear function.

3.1.2 Verification and Validation of Surrogate Models

The process of generating a data set from code simulations and constructing a surrogate model through regression analysis is known as surrogate training. The data set is referred to as the training set. Before the code surrogate can be applied to an engineering application, the surrogate must be validated and verified. The required performance of a surrogate will be application specific but in general, the adequacy is measured by the capability of the surrogate to accurately predict the output response of interest for a range of input parameter values within an acceptable uncertainty or error to the original code model.

In practice, when constructing a code surrogate to approximate a complex engineering computer code such as a TH system code, the input parameter space will be large and very little *a priori* information about the possibly highly nonlinear functional relationship will be known. For this type of application, a sufficiently complex and flexible parametric model which could run a high risk of over fitting or a nonparametric model would have to be used. If the latter

option is chosen the training set must be sufficiently large. Even if *a priori* information about the functional form is known and a parametric model can be used, the model will likely have to incorporate many basis functions to capture the interaction terms between input parameters. The training set would still need to be large enough in order to calculate the large number of fitting coefficients. Regardless of regression method, the training set must also adequately cover the large input parameter space. Numerical strategies referred to as experimental design such as stratified and Latin hypercube sampling (LHS) [McK79] and space filling designs [Joh90] exist for selecting input values for the training set.

For a desired accuracy of the surrogate, there is some optimal set of data points required for training that represents the lower bound on the size of the training set. The training set can only be generated by running expensive code simulations so the upper bound on size of the set must always be finite and limited by the computational resources available. Ultimately, the decision to use a code surrogate in place of the original code model is a subjective process using engineering judgment to interpret a variety of quantitative tests such as goodness of fit and resampling techniques used to assess the surrogate model constrained by the finite training set size.

The goodness of fit is a measurement of how well a surrogate model fits to the data in training set. The surrogate can be evaluated at each set of input values from the training set and the residual error calculated for each prediction $\hat{y}_i$ from which the coefficient of determination or R^2 value can be calculated. The coefficient of determination is calculated as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad , \tag{3.6}$$

with $\bar{y}$ equal to the sample mean of the outputs in the training set. R^2 values close to 1 indicates close agreement between the surrogate model and the observations in the training set. While the coefficient of determination is a useful statistic for goodness of fit, there are subtleties to consider when interpreting R^2 values of code surrogates. To illustrate this point, we will consider two simple linear regression problems applied to points drawn from a linear function $y = x + \varepsilon$ with ε distributed as $U(-1,1)$ and an exponential function $y = e^x$. Figure 3.2 shows plots of 1st degree polynomial linear regression models fitted to data both sets of data points and an additional 15th and 3rd degree polynomial linear model fitted to the linear function data set and

exponential function data set, respectively. For the linear function data set, the 1st degree polynomial fit yielded a R^2 value of 0.23 suggesting a poor fit to the noisy data. However, the equation of the fit is $y = 0.9177x + 1.49\times10^{-4}$ which is a very accurate estimate of the true underlying function. The small R^2 value is a result of the large variance of the sampled data. The 15th degree polynomial fit yielded a R^2 value of 0.88 suggesting a good fit. Inspection of Figure 3.2 shows the highly oscillatory behavior of the 15th degree polynomial fit which is a clear over fitting of the data by an overly complex surrogate model. The 1st degree polynomial model fitted to the exponential function yielded a R^2 value of 0.79 suggesting an acceptable fit to the data but through visual inspection of the fit and knowledge of the underlying exponential function, we know a 1st degree polynomial is not a very good approximation to an exponential function. The Taylor series expansion of the exponential function is the infinite sum

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad , \tag{3.7}$$

so it is not surprising the 3rd order polynomial fit is a very good approximation yielding a R^2 value of 0.999 because a 3rd order polynomial matches the functional form of the first four moments of the Taylor series expansion.

Important conclusions can be drawn from this simple example. The calculated R^2 value alone is not enough information to conclude whether or not a surrogate model is an adequate approximation to the underlying function. Over fitting the surrogate model to the data needs to be avoided. Over fitted surrogates can yield an R^2 value close to 1 for the training set data but predictive accuracy at new points in the input parameter space will often be very poor. If information about the underlying functional form of the input/output relationship is known, a very accurate surrogate can be trained from the data using parametric regression analysis.

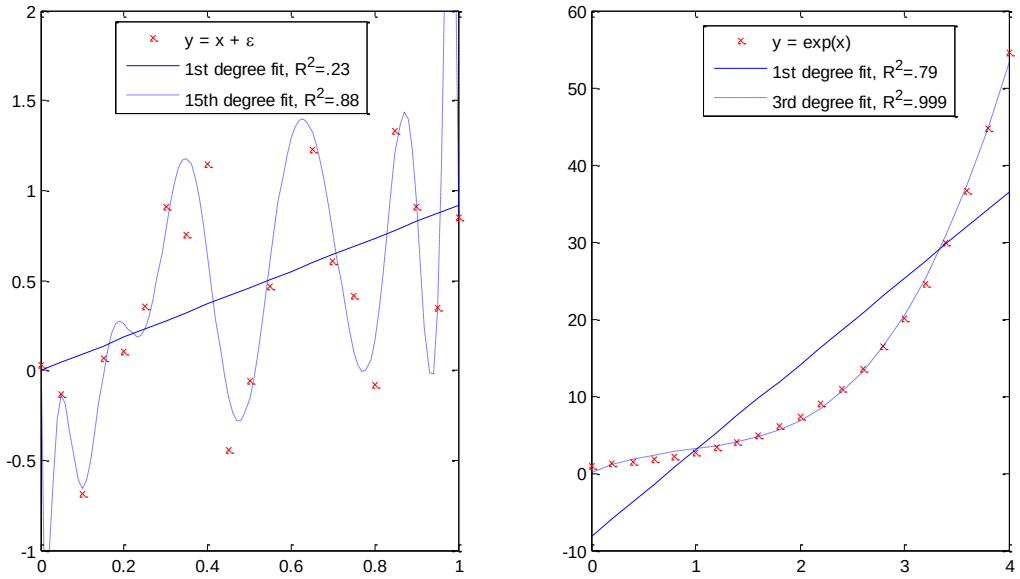

Figure 3.2. Linear regression models fitted to data drawn from $y = x$ with noise and $y = e^x$.

When dealing with a finite sample of data, extra information can be extracted through careful manipulation of the data set and analyzing subsets of the complete sample through resampling techniques, namely cross-validation, jackknifing, and bootstrapping methods. Cross-validation purposely leaves a subset of data, the test set, out of the training set. To test the predictive accuracy of the surrogate, the surrogate predictions at the test set of inputs are compared with the output response variables in the test set. Statistical properties of the error residuals give information about the accuracy and bias of the surrogate. The surrogate predictions should meet the desired accuracy required and be unbiased or else additional data points need to be added to the training set or a different functional form chosen for the surrogate if the surrogate is a parametric model. Bias in the surrogate is generally undesirable but in some nuclear safety applications, bias is acceptable as long as the model is conservative.

Similar to cross-validation, jackknifing removes a subset of data from the training set. The surrogate is retrained on the remaining data. The process is repeated for different subsets from training set and the statistics from each resulting surrogate can be compared. For a training set that is sufficiently large and surrogate functional form that is complex and flexible enough to model the underlying functional relationship, the performance of the surrogate trained on the

complete training set and each jackknife sample should be comparable and any surrogate model coefficients and parameters should not fluctuate significantly in value across the different data sets. Fluctuating model parameter values indicate the fitted surrogate is sensitive to individual or subsets of data points in the training set either from over fitting or a training set that is too small to capture all of the underlying functional relationships so each data point provides a new or unique piece of information for surrogate training.

Bootstrapping does not remove data points from the training set but rather resamples by adding randomly generated error residuals to the output response variables in the training set. The surrogate is retrained on the bootstrapped data set and variation between the models are observed. Bootstrapping has the benefit of retaining all of the test data, which may be very expensive to obtain, in the training set.

3.2 Gaussian Process Model

The GPM is unique among regression methods because it defines a *predictive distribution* $p(f_*|\boldsymbol{x}_*,X,\boldsymbol{y})$ of the output f_* at any input $\boldsymbol{x}_* = \{x_i, i=1,\ldots,p\}$ given a training set of n data points $(X,\boldsymbol{y})=\{\boldsymbol{x}_j, y_j, j=1,\ldots,n\}$. The predictive distribution is a Gaussian $N(\bar{f}_*, V_*)$ defined by a mean function and variance:

$$\bar{f}_* = \boldsymbol{k}_*^T\left(K + V_y I\right)^{-1}\boldsymbol{y} \tag{3.8}$$

$$V\left[\bar{f}_*\right] = V_* = k(\boldsymbol{x}_*,\boldsymbol{x}_*) - \boldsymbol{k}_*^T\left(K + V_y I\right)^{-1}\boldsymbol{k}_* \tag{3.9}$$

$$K = \begin{bmatrix} k(\boldsymbol{x}_1,\boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_1,\boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n,\boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_n,\boldsymbol{x}_n) \end{bmatrix} \tag{3.10}$$

$$\boldsymbol{k}_* = [k(\boldsymbol{x}_*,\boldsymbol{x}_1);\ k(\boldsymbol{x}_*,\boldsymbol{x}_2);\ldots;k(\boldsymbol{x}_*,\boldsymbol{x}_n)]\,. \tag{3.11}$$

Here the kernel or covariance function $k(\boldsymbol{x}_q, \boldsymbol{x}_r) = cov[f(\boldsymbol{x}_q), f(\boldsymbol{x}_r)]$ has been introduced which defines the covariance between any two data points. I is the identity matrix. We will formally derive Eqs. (3.8) and (3.9) following the derivation given in [Ras06].

We begin with the linear regression model discussed in section 3.1.1 assuming a simple linear polynomial fit and replacing the notation of fitting coefficients β with regression weights $\boldsymbol{w}$

$$f(X) = X^T \boldsymbol{w}, \qquad \boldsymbol{y} = f(X) + \varepsilon \quad , \tag{3.12}$$

where noise is distributed as a zero mean Gaussian with variance V_y

$$p(\varepsilon) = N(0, V_y) \, . \tag{3.13}$$

Given the model, the likelihood of observing the data y is the joint PDF

$$p(\boldsymbol{y}|X, \boldsymbol{w}) = \frac{1}{\sqrt{2\pi V_y}} \exp\left(-\frac{1}{2V_y}|\boldsymbol{y} - X^T \boldsymbol{w}|^2\right) = N(X^T \boldsymbol{w}, V_y I) \quad . \tag{3.14}$$

Unless prior constraints are applied, Eq. (3.14) will hold for any set of weights recognizing there are an infinite number possible weight values. An arbitrarily chosen set of weights will most likely yield a low likelihood probability while a set derived from the data by ordinary least squares given by Eq. (3.4), will yield a higher likelihood. Through Bayes' theorem, the posterior distribution of the weights given the data can be computed by

$$p(\boldsymbol{w}|\boldsymbol{y}, X) = \frac{p(\boldsymbol{y}|X, \boldsymbol{w}) p(\boldsymbol{w})}{p(\boldsymbol{y}|X)} \tag{3.15}$$

with the prior distribution of the weights assumed to be a zero mean Gaussian with covariance matrix V_w

$$p(\boldsymbol{w}) = N(0, V_w) \quad . \tag{3.16}$$

The marginal likelihood $p(\boldsymbol{y}|X)$ is a normalization constant so Eq. (3.15) can be expressed as

$$p(\boldsymbol{w}|\boldsymbol{y},X) \propto \exp\left(-\frac{1}{2V_y}(\boldsymbol{y}-X^T\boldsymbol{w})^T(\boldsymbol{y}-X^T\boldsymbol{w})\right)\exp\left(-\frac{1}{2}\boldsymbol{w}^T V_w^{-\mathbf{1}}\boldsymbol{w}\right)$$
$$\propto \exp\left(-\tfrac{1}{2}(\boldsymbol{w}-\bar{\boldsymbol{w}})^T\left(V_y^{-1}XX^T+V_w^{-1}\right)(\boldsymbol{w}-\bar{\boldsymbol{w}})\right), \quad (3.17)$$

with $\bar{\boldsymbol{w}} = V_y^{-1}VX\boldsymbol{y}$ and $V^{-1} = \left(V_y^{-1}XX^T + V_w^{-1}\right)$. Equation (3.17) has the form of a Gaussian with mean $\bar{\boldsymbol{w}}$ and covariance matrix *V*. Next we will explicitly derive Eq. (3.17). In order to avoid the complicated vector and matrix notation, we consider a simplified one-dimensional case where $f(x) = wx$ and a single measurement *y* has been taken. The posterior distribution for *w* is

$$p(w|y,X) \propto \exp\left(-\frac{1}{2V_y}(y-Xw)(y-Xw)\right)\exp\left(-\frac{1}{2}wV_w^{-1}w\right)$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{y^2}{V_y}-\frac{2yXw}{V_y}+\frac{X^2w^2}{V_y}+V_w^{-1}w^2\right)\right)$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{y^2}{V_y}-\frac{2yXw}{V_y}+\left(\frac{X^2}{V_y}+V_w^{-1}\right)w^2\right)\right). \quad (3.18)$$

We are seeking a distribution for *w* that has a form of a Gaussian so we rearrange Eq. (3.18)

$$p(w|y,X) \propto \exp\left(-\frac{1}{2\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\left(\frac{y^2}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}-\frac{2yXw}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}+w^2\right)\right). \quad (3.19)$$

Completing the square with the quadratic expression involving *w*, we obtain:

$$\propto \exp\left(-\frac{1}{2\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\left(\frac{y^2}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}-\left(\frac{yX}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\right)^2+\left(w-\frac{Xy}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\right)^2\right)\right)$$

$$\propto \exp\left(-\frac{1}{2\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\left(\frac{y^2}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}-\left(\frac{yX}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\right)^2\right)\right)\exp\left(-\frac{1}{2\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\left(w-\frac{Xy}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\right)^2\right). \quad (3.20)$$

The first exponential term is only dependent on constant values so it evaluates to a constant and can be included within the proportionality constant of the distribution and Eq. (3.20) simplifies to

$$p(w|y,X) \propto \exp\left(-\frac{1}{2\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\left(w-\frac{Xy}{V_y\left(\frac{X^2}{V_y}+V_w^{-1}\right)}\right)^2\right) \quad , \tag{3.21}$$

which we recognize as a Gaussian with mean $\bar{w} = V_y\left(V_y^{-1}X^2 + V_w^{-1}\right)^{-1}Xy$ and variance $\left(V_y^{-1}X^2 + V_w^{-1}\right)$ and is equivalent to Eq. (3.17).

To obtain the predictive distribution $p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y})$, an average over all possible models defined by weights $\boldsymbol{w}$ must be taken weighted by the posterior distribution of the weights given by Eq. (3.17). First, $p(f_*|\boldsymbol{x}_*, \boldsymbol{w})$, the probability of obtaining the prediction f_* given a model with weights $\boldsymbol{w}$ and input $\boldsymbol{x}_*$, must be defined. A delta function is an appropriate form for $p(f_*|\boldsymbol{x}_*, \boldsymbol{w})$ because an unique set of weights and single input will yield an unique output. The predictive distribution is obtained by the integral

$$\begin{aligned} p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) &= \int p(f_*|\boldsymbol{x}_*, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{y}, X)d\boldsymbol{w} \\ &= \int \delta(f_* - \boldsymbol{x}_*^T\boldsymbol{w})N(\bar{\boldsymbol{w}}, V)\,d\boldsymbol{w} \\ &= N(\boldsymbol{x}_*^T\bar{\boldsymbol{w}}, \boldsymbol{x}_*^T V\boldsymbol{x}_*) \ . \end{aligned} \tag{3.22}$$

Equation (3.22) was derived with the initial assumption of the simple linear polynomial fit of Eq. (3.12). Of course linear regression models are not limited to linear fits and any set of basis functions $\phi(\boldsymbol{x}) = \{\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_m(\boldsymbol{x})\}^T$ can be incorporated into the model replacing $\boldsymbol{x}$ with $\phi(\boldsymbol{x})$ and X with $\Phi(X)$ where the columns of Φ are $\phi(\boldsymbol{x})$ for $\boldsymbol{x}$ in the training set. After substitutions, Eq. (3.22) is

$$p(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = N(\phi_*^T\bar{\boldsymbol{w}}, \phi_*^T V\phi_*) \ , \tag{3.23}$$

with the shorthand notation $\phi(\boldsymbol{x}_*) = \phi_*$ introduced. The mean weights and covariance matrix are $\bar{\boldsymbol{w}} = V_y^{-1} V \Phi \boldsymbol{y}$ and $V^{-1} = \left(V_y^{-1} \Phi \Phi^T + V_w^{-1}\right)$, respectively. Through a series of matrix operations given in [Ras06], Eq. (3.23) can be expressed as

$$(f_*|\boldsymbol{x}_*, X, \boldsymbol{y}) = N(\phi_*^T V_w \Phi \left(\Phi^T V_w \Phi + V_y I\right)^{-1} \boldsymbol{y}, \\ \phi_*^T V_w \phi_* - \phi_*^T V_w \Phi \left(\Phi^T V_w \Phi + V_y I\right)^{-1} \Phi^T V_w \phi_*) \ . \tag{3.24}$$

Taking a closer look at Eq. (3.24), input variables always appear in vector and matrix operations with the covariance matrix V_w as $\phi_*^T V_w \phi_*$, $\phi_*^T V_w \Phi$, $\Phi^T V_w \phi_*$, or $\Phi^T V_w \Phi$. Since the columns of Φ are ϕ, every matrix operation is comprised of the inner products $\phi(\boldsymbol{x}_q)^T V_w \phi(\boldsymbol{x}_r)$ where $\boldsymbol{x}_q$ and $\boldsymbol{x}_r$ are either part of the training set or the prediction point. A new variable, the covariance function, can be defined $k(\boldsymbol{x}_q, \boldsymbol{x}_r) = \phi(\boldsymbol{x}_q)^T V_w \phi(\boldsymbol{x}_r)$. From the definition for $k(\boldsymbol{x}_q, \boldsymbol{x}_r)$, Eqs. (3.8) and (3.9) are equivalent to Eq. (3.24) recognizing $K = \Phi^T V_w \Phi$ and $\boldsymbol{k}_* = \phi_*^T V_w \Phi = \Phi^T V_w \phi_*$.

3.2.1 GPM Covariance Function

The covariance function $k(\boldsymbol{x}_q, \boldsymbol{x}_r)$ is the most important element of the GPM because it encodes all information about the underlying function we are trying to infer from the training data. The basic assumption applied to the selection of $k(\boldsymbol{x}_q, \boldsymbol{x}_r)$ is that data points with inputs that are close in the input parameter space are likely to have correlated outputs; therefore, $k(\boldsymbol{x}_q, \boldsymbol{x}_r)$ should define the covariance between two outputs $cov[f(\boldsymbol{x}_q), f(\boldsymbol{x}_r)]$ based on the input data locations. A common covariance function is the squared exponential function

$$k(\boldsymbol{x}_q, \boldsymbol{x}_r) = \theta_1 \exp\left(-\tfrac{1}{2}(\boldsymbol{x}_q - \boldsymbol{x}_r)^T \boldsymbol{\Lambda}^{-1} (\boldsymbol{x}_q - \boldsymbol{x}_r)\right) + \theta_2, \tag{3.25}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & r_p \end{bmatrix} \ . \tag{3.26}$$

The constant θ_1 is a scaling factor and each r_i is the characteristic length scale of the process in the dimension *i* of the input parameter space. The second term θ_2 in the covariance function is a constant offset which allows the underlying function to have nonzero mean. These factors are generally referred to as hyperparameters. The squared exponential covariance function is widely used in GPMs and has performed well in previous analysis [Bai99,Ras06].

The squared exponential covariance function assigns covariances through a parameterized (by hyperparameters θ and $r_{i,i=1,\ldots,p}$) distance-based metric. The values of the hyperparameters must be inferred or learned from the data by maximizing the likelihood of the hyperparameters given the data by gradient search methods [Ras06] through a procedure referred to as Automatic Relevance Determination (ARD) [Nea96]. The squared exponential covariance function is amenable to ARD because the inverse of each length scale determines the sensitivity of the output to the input dimension. If an r_i is large, the covariance is independent of the *i*th input dimension.

3.2.2 Interpreting the GPM

The GPM is the mean and variance functions of Eqs. (3.8) and (3.9) comprised of matrix and vector operations. The entries of the $n \times n$ matrix K are the covariances between the training set data points with the measurement noise added to the diagonal entries. The first matrix-vector multiplication of Eq. (3.8) is $\left(K + V_y I\right)^{-1} \boldsymbol{y}$ which yields a new $n \times 1$ vector $\boldsymbol{y}_s$ of "smoothed" output data values for each training point. The inner product of the *i*th row and $\boldsymbol{y}$ yields a weighted average of the output $\boldsymbol{y}$ for the *i*th data point with the weights determined by the covariance between the *i*th point and each of the other data points in the training set. The weighted average is the smoothing operation. The matrix inversion and multiplication with $\boldsymbol{y}$ only has to be performed once and the result stored electronically. The final operation to calculate the prediction mean at the prediction point $\boldsymbol{x}_*$ is a second smoothing operation of $\boldsymbol{k}_*^T \boldsymbol{y}_S$. The weighted average of the vector of smoothed output data is taken with the weights determined by the covariance between $\boldsymbol{x}_*$ and each of the data points in the training set. After the initial matrix inversion, calculating the prediction mean requires order (*n*) operations for each

prediction point. Similarly for Eq. (3.9), the matrix inversion only has to be performed once and each prediction variance calculation requires order (n^2) operations.

In a classical response surface formulation, the prediction mean function could be interpreted as the input/output mapping and the prediction variance function would a quantitative measure of the model uncertainty. However, the GPM is not trying to approximate the functional relationship between the input and output parameters in the traditional data fitting sense but rather is describing a predictive distribution at any location in the input parameter space which is a Gaussian defined by the prediction mean and variance. The GPM is often described as a distribution over functions implying there are many (possibly infinite) functions that could be the true underlying function with certain functions having a higher likelihood given the training set. The prediction mean is the most *probable* output value an arbitrary function selected from the complete set of possible functions would produce at that particular location in the input parameter space, noting that the GPM is not making any inference about the function behavior anywhere else in the input parameter space other than at the prediction point. The prediction variance constrains the range of probable values a function can take about the mean.

3.3 Alternating Conditional Expectation Algorithm

The ACE algorithm [Bre85] yields an optimal relationship between the dependent variable and multiple independent variables by obtaining one-dimensional transformations $\theta(y)$ and $\phi_i(x_i)$ of each variable through an iterative procedure that maximizes the statistical correlation between $\theta(y)$ and $\sum_{i=1}^{p} \phi_i(x_i)$. The transformations satisfy the linear relationship

$$\theta(y) = \sum_{i=1}^{p} \phi_i(x_i) + \varepsilon \quad . \tag{3.27}$$

The transformations are obtained by minimizing the square error of a linear relationship between the transformed dependent variable $\theta(y)$ and the sum of the transformed independent variables $\sum_{i=1}^{p} \phi_i(x_i)$

$$\varepsilon^2(\theta, \phi_1, \ldots, \phi_p) = E\left[\theta(y) - \sum_{i=1}^{p} \phi_i(x_i)\right]^2 \quad . \tag{3.28}$$

The minimization of ε^2 with respect to each transformed variable yields

$$\phi_i(x_i) = E\left[\theta(y) - \sum_{j\neq i}^{p} \phi_j(x_j) \,|\, x_i\right] \quad , \tag{3.29}$$

$$\theta(y) = \frac{E\left[\sum_{i=1}^{p} \phi_i(x_i)|y\right]}{\left\|E\left[\sum_{i=1}^{p} \phi_i(x_i)|y\right]\right\|} \quad , \tag{3.30}$$

with the square-norm $\|\cdot\|$ introduced such that $E[\theta^2(y)]=1$. The ACE algorithm iteratively solves Eqs. (3.29) and (3.30) to converge to an estimate of the optimal transformations $\theta^*(y)$ and $\phi_i^*(x_i)$. The derivation of the ACE algorithm given in [Bre85] provides a mathematical proof of the existence of the optimal transformations $\theta^*(y)$ and $\phi_i^*(x_i)$. Given the training set of n data points $(X,\mathbf{y})=\{x_{ij},y_j,\ i=1,..,p,\ j=1,\ldots,n\}$, the ACE algorithm must compute a set of transformed data points $\{\phi_i(x_{ij}),\ \theta(y_j)\}$ that are realizations of the continuous transformations $\theta(y)$ and $\phi_i(x_i)$. The ACE algorithm is

1) Initialize $\theta(y) = y/\|y\|$ and all $\phi_i(x_i) = 0$.

2) Calculate $\phi_i(x_i)$ conditioned on x_i. Sort $\theta(y)$ and $\phi_l(x_l)$ in ascending order of x_i and evaluate for $i = 1,..,p$:

$$\phi_i(x_i) = E\left[\theta(y) - \sum_{l\neq i}^{p} \phi_l(x_l) \,|\, x_i\right]$$

   Iterate through all i until squared error fails to decrease

$$\varepsilon^2(\theta, \phi_1, \ldots, \phi_p) = \frac{1}{n}\sum_{j=1}^{n}\left[\theta(y_j) - \sum_{i=1}^{p} \phi_i(x_{ij})\right]^2$$

   All $\theta(y)$ are held constant and $\phi_i(x_i)$ is updated after each iteration.

3) Calculate $\theta(y)$ conditioned on y. Sort $\phi_i(x_i)$ in ascending order of y and evaluate:

$$\theta(y) = \frac{E\left[\sum_{i=1}^{p} \phi_i(x_i)|y\right]}{\left\|E\left[\sum_{i=1}^{p} \phi_i(x_i)|y\right]\right\|}$$

4) Alternate between steps 2 and 3 until $\varepsilon^2(\theta, \phi_1, \ldots, \phi_p)$ does not change.

In steps 2 and 3, the individual transformed data points $\phi_i(x_{ij})$ and $\theta(y_j)$ are calculated by a localized smoothing operation about the jth point

$$\begin{aligned}\phi_i(x_{ij}) &= E\left[\theta(y) - \textstyle\sum_{l\neq i}^{p} \phi_l(x_l) \,|x_i = x_{ij}\right] \\ &= S\left[\theta(y_k) - \textstyle\sum_{l\neq i}^{p} \phi_l(x_{lk}) \,|x_{ij}\right] \quad ,\end{aligned} \tag{3.31}$$

$$\theta(y_j) = \frac{E\left[\sum_{i=1}^{p} \phi_i(x_i)|y=y_j\right]}{\left\|E\left[\sum_{i=1}^{p} \phi_i(x_i)|y=y_j\right]\right\|} = \frac{S\left[\sum_{i=1}^{p} \phi_i(x_{ik})|y_j\right]}{\left\|S\left[\sum_{i=1}^{p} \phi_i(x_{ik})|y_j\right]\right\|} \quad , \tag{3.32}$$

$$S[Z_k|z_j] = \textstyle\sum_{k=j-M}^{j+M} W_k Z_k \ . \tag{3.33}$$

The smoothing operation defined in Eq. (3.33) takes the form of a weighted average about the window of data points. The weights W_k and window width $2M$ are determined by the type of smoothing operation which must be chosen by the user. The supersmoother [Fri82] which allows the window width to vary is implemented in the ACE algorithm presented in [Bre85] and in the 'acepack' package available in the R program. [Kim97] implemented the tri-cube function [Cle79]. The smoothing operation is the conditional expectation from which the name for the ACE algorithm is derived.

The ACE algorithm is a very powerful nonparametric regression technique. ACE guarantees convergence without assumptions about the underlying functional forms of the transformations. After the iterations have converged, each transformed set of points $\{x_i, \phi_i(x_i)\}$ and $\{y,\theta(y)\}$ are slowly varying and the multi-dimensional input space $\boldsymbol{x}$ is mapped to the single dimension transform space. Likewise, the output dimension y is mapped to the same transform space. The global variation of y to the transform space is represented by $\theta(y)$ and each $\phi_i(x_i)$ represents the local variation in the transformed space by variation in the input parameter x_i. Through the common transform space, the nonlinear mapping $\boldsymbol{x} \rightarrow \{\Sigma\phi,\theta\} \rightarrow y$ is achieved. Each value of ϕ_i at a prediction point is obtained using transformations as interpolation tables and the corresponding θ is calculated using Eq. (3.27). The final step in the nonlinear mapping involves

taking the inverse transform $\theta^{-1}\left(\sum_{i=1}^{p}\phi_i(x_i)\right)$ to obtain y. The one-dimensional transformations can be visually inspected giving physical insights into the input-output relationship of the variables.

3.3.1 Variance Estimate of ACE Transforms

Both the GPM and ACE are nonparametric regression techniques that use data smoothing as the primary tool for surrogate training. ACE performs data smooths on the transformed data points for each input dimension and the output dimension and the GPM performs a data smooth directly on the output values in the training set. A subtle difference between the methods is that the ACE transformations represent the best estimate of the true underlying functional relationship whereas the GPM is a distribution over functions as discussed in Section 3.2.2. The distinct advantage of the GPM is the prediction variance which quantifies the model uncertainty of the GPM. Drawing on the similarities of methods, we propose a prediction variance formulation for the ACE algorithm.

In the last iteration of the ACE algorithm, the final converged transformed dependent variable points $\theta(y_j)$ are obtained through the data smooth given by Eqs. (3.32) and (3.33)

$$\theta(y_j) = \sum_{i=1}^{p}\left[\sum_{k=j-M}^{j+M} W_{ik}\phi_i(x_{ik})\right] . \tag{3.34}$$

The inner summation in Eq. (3.34) is a weighted average over the subset of $2M$ transformed data points in the transformed dimension i yielding a weighted mean of $\phi_i|y_j$ to be used in the calculation of $\theta(y_j)$. Similarly, the weighted variance of the subset can be calculated. The weighted mean and variance for dimension i is

$$E[\phi_i|y_j] = \bar{\phi}_{ij} = \sum_{k=j-M}^{j+M} W_{ik}\phi_i(x_{ik}) \quad \text{and} \tag{3.35}$$

$$V\left[\phi_i|y_j\right] = s_{ij}^2 = \frac{\sum_{k=j-M}^{j+M} W_{ik}\left(\phi_i(x_{ik})-\bar{\phi}_{ij}\right)^2}{\frac{2M-1}{2M}\sum_{k=j-M}^{j+M} W_{ik}} \quad . \tag{3.36}$$

Next we use Eqs. (3.35) and (3.36) to assume $p(\phi_i|y_j)$ is distributed as a Gaussian $N(\bar{\phi}_{ij}, s_{ij}^2)$. Each $\theta(y_j)$ is now a summation of p Gaussian random variables so $\theta|\Sigma\phi_i$ is best described as a PDF which is a Gaussian

$$P(\theta|\sum\phi_i) = \sum_{i=1}^{p} p(\phi_i|y_j) = \sum_{i=1}^{p} N(\bar{\phi}_{ij}, s_{ij}^2)$$

$$= N\left(\sum_{i=1}^{p}\bar{\phi}_{ij},\ \sum_{i=1}^{p} s_{ij}^2\right)$$

$$= N\left(\sum_{i=1}^{p}\left[\sum_{k=j-M}^{j+M} W_{ik}\phi_i(x_{ik})\right],\ \sum_{i=1}^{p} s_{ij}^2\right)$$

$$P(\theta|\sum\phi_i) = N\left(\theta(y_j), V[\theta(y_j)]\right) \quad . \tag{3.37}$$

Equation (3.37) indicates the mean of $\theta|\Sigma\phi_i$ is $\theta(y_j)$ obtained from the last converged iteration of ACE and the variance is the summation of the weighted variances given by Eq. (3.36).

The training set is now represented as a set of transformed data points $\{\phi_i(x_{ij}),\ \theta(y_j),\ V[\theta(y_j)]\}$. The uncertainty of the ACE model appears in the θ component of the mapping of $\Sigma\phi_i \rightarrow \theta$ in the transform space. The uncertainty is derived from the sample statistics of the subsets of transformed data used in the data smoothing operations of the ACE algorithm. As a code surrogate, the ACE model can be expressed with model uncertainty added as model noise v which is distributed as $N(0, V[\theta(y_j) = \Sigma\phi_i])$

$$\theta(y) = \sum_{i=1}^{p}\phi_i(x_i) + v \quad . \tag{3.38}$$

As in the GPM, we want to know the uncertainty in a prediction y given the uncertainty of the surrogate model. For the ACE model, this means the uncertainty in θ must be propagated to y involving the nontrivial transform of the Gaussian distribution of v through the possibly nonlinear inverse transformation $\theta^{-1}(\ )$

$$y = \theta^{-1}\left(\sum_{i=1}^{p}\phi_i(x_i), v\right) \quad . \tag{3.39}$$

The unscented transformation presented in the following sections is the method proposed to evaluate Eq. (3.39) and obtain estimates of the mean and variance of y subject to the model uncertainty of the ACE model.

3.4 Unscented Transform

Nonparametric regression techniques, the GPM and ACE algorithm, were presented in Sections 3.2 and 3.3 as alternatives to conventional linear regression models. We present a new deterministic sampling method, the unscented transform (UT) to complement random sampling methods such as Wilks' formula in direct sampling approaches for safety analysis for NPPs. The UT is a general method for propagating distributions through nonlinear functions.

The UT was developed by Julier and Uhlmann [Jul04] in the context of extending the Kalman Filter (KF) to nonlinear system dynamics. The KF is one of the most widely used predictor-corrector algorithms used to estimate the mean and covariance of system states of dynamic systems described by linear process and observation models subject to noise. The KF would break down when applied to nonlinear systems so there was a need to accurately predict statistical properties of random variables transformed through nonlinear functions.

Consider a random variable $\boldsymbol{x}$ that is a $n \times 1$ vector defined by a mean vector $\bar{\boldsymbol{x}}$ and covariance $\boldsymbol{P}$ that is related to the output variable y by the nonlinear function

$$y = f(\boldsymbol{x}) \quad . \tag{3.40}$$

The statistics of y are dependent on the propagation of the statistical properties of $\boldsymbol{x}$ through $f()$. Although it is difficult or impossible to transform a PDF through a nonlinear function, nonlinear transforms of individual points are easy to perform. The foundation of the UT is that if a set of m carefully chosen points, called sigma points, and associated weighting coefficients $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}, W^{(i)}: i = 0,1,...,m\}$ whose sample mean and covariance is equal to $\bar{\boldsymbol{x}}$ and $\boldsymbol{P}$, are passed through $f()$, the statistics of the transformed points which are realizations of the output response y, contain high order accuracy information about the PDF of y. The weights can be positive or negative but are constrained by

$$\sum_{i=0}^{m} W^{(i)} = 1 \qquad . \tag{3.41}$$

One generalized set of $m = 2n+1$ sigma points $\boldsymbol{x}^{(i)}$ satisfying the criteria for the UT is

$$\begin{aligned} \boldsymbol{x}^{(0)} &= \bar{\boldsymbol{x}} \\ \boldsymbol{x}^{(i)} &= \bar{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^{(i)} \quad i = 1, \ldots, 2n \\ \tilde{\boldsymbol{x}}^{(i)} &= \left(\sqrt{(n+k)\boldsymbol{P}}\right)_i \quad i = 1, \ldots, n \\ \tilde{\boldsymbol{x}}^{(n+i)} &= -\left(\sqrt{(n+k)\boldsymbol{P}}\right)_i \quad i = 1, \ldots, n \; . \end{aligned} \tag{3.42}$$

The $2n+1$ weighting coefficients are

$$W^{(0)} = \frac{k}{n+k}$$

$$W^{(i)} = \frac{1}{2(n+k)} \quad i = 1, \ldots, 2n \; . \tag{3.43}$$

The unscented estimate of the mean and covariance of y are calculated from the transformed sigma points $y^{(i)}$

$$\begin{aligned} y^{(i)} &= f\left(\boldsymbol{x}^{(i)}\right) \\ \bar{y}_u &= \sum_{i=0}^{2n} W^{(i)} y^{(i)} \end{aligned}$$

$$P_u = \sum_{i=0}^{2n} W^{(i)} \left(y^{(i)} - \bar{y}_u\right)\left(y^{(i)} - \bar{y}_u\right)^T \; . \tag{3.44}$$

The UT mean and covariance $\bar{y}_u$ and P_u are accurate up to the 3rd order if the distribution of $\boldsymbol{x}$ is symmetric [Jul04,Sim06]. The variable k in Eqs. (3.42) and (3.43) is a free parameter that can take on any value as long as $(n + k) \neq 0$. However, certain values of k can reduce the error in the 4th and higher-order terms dependent on the type of input PDFs. If $\boldsymbol{x}$ is Gaussian then $k = 3 - n$ will minimize error in the 4th order terms [Jul04,Sim06]. If $k = 0$, then a zero weight is assigned to the central sigma point $\boldsymbol{x}^{(0)}$ and the sigma set is reduced to $2n$ symmetric points.

Equations (3.42) through (3.44) are the UT. A mathematical subtlety that must be considered is the calculation of the matrix square root $\boldsymbol{A} = \sqrt{(n+k)\boldsymbol{P}}$ that appears in the definition of the sigma points. If the matrix square root $\boldsymbol{A}$ of $(n + k)\boldsymbol{P}$ has the form $(n + k)\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ then $\left(\sqrt{(n+k)\boldsymbol{P}}\right)_i$ designates the i^{th} column of $\boldsymbol{A}$. If it has the form $(n + k)\boldsymbol{P} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, then the transpose of the rows of $\boldsymbol{A}$ are taken. If the variables of $\boldsymbol{x}$ are independent of each other, the covariance matrix $\boldsymbol{P}$ will be a diagonal with the diagonal entries equal to the variance of each dimension of $\boldsymbol{x}$. A diagonal matrix $\boldsymbol{P}$ with real entries has a unique matrix square root $\boldsymbol{A}_u$ which is simply another diagonal matrix with diagonal entries equal to the square root of the diagonal entries of $\boldsymbol{P}$. However, for a matrix $\boldsymbol{P}$, there are an infinite number of matrix square roots considering a new matrix square root $\boldsymbol{A}_2 = \boldsymbol{A}\mathbf{Q}$ can be defined where $\mathbf{Q}$ is any orthogonal matrix that is the same size of $\boldsymbol{A}$. $\mathbf{Q}$ performs an orthogonal transformation of $\boldsymbol{A}$. From the properties of matrix multiplication, $\boldsymbol{P} = \boldsymbol{A}_2\boldsymbol{A}_2^T = (\boldsymbol{A}\boldsymbol{Q})(\boldsymbol{A}\boldsymbol{Q})^T = \boldsymbol{A}\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}^T = \boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{P}$. For the UT, any matrix square root can be used [Jul04].

Considering a nuclear safety calculation where a safety parameter such as peak clad temperature (PCT) is the output variable y of interest, the TH computer code used to simulate the transient can be interpreted as a complex nonlinear function with many uncertain input parameters. For most TH code calculations, the input parameters will be independent so the covariance matrix $\boldsymbol{P}$ will be diagonal and the unique matrix square root $\boldsymbol{A}_u$ can be readily calculated. If the UT is to be used to estimate the mean and variance of the PCT and $\boldsymbol{A}_u$ is the specific matrix square root used in Eq. (3.42), then the resulting sigma point set will have an undesirable property. Each column of $\boldsymbol{A}_u$ has only one non-zero element so only one input dimension will be modified for each sigma point leaving the other n - 1 dimensions held at the mean values. Modifying one variable at a time while holding all others constant is analogous to a sensitivity calculation. A sigma point set with this property would not be expected to account for combined effects of simultaneous variation of input parameters resulting from interaction terms or coupling of variables in the nonlinear function. A more robust sigma set would be obtained by transforming $\boldsymbol{A}_u$ by an orthogonal matrix [Zan01]. A randomly generated orthogonal matrix could ensure that all elements of $\tilde{\boldsymbol{x}}^{(i)}$ would be nonzero and all input parameters would be varied for each sigma point except for the central point $\boldsymbol{x}^{(0)}$. Figure 3.3 illustrates an orthogonal transform of a two variable sigma point.

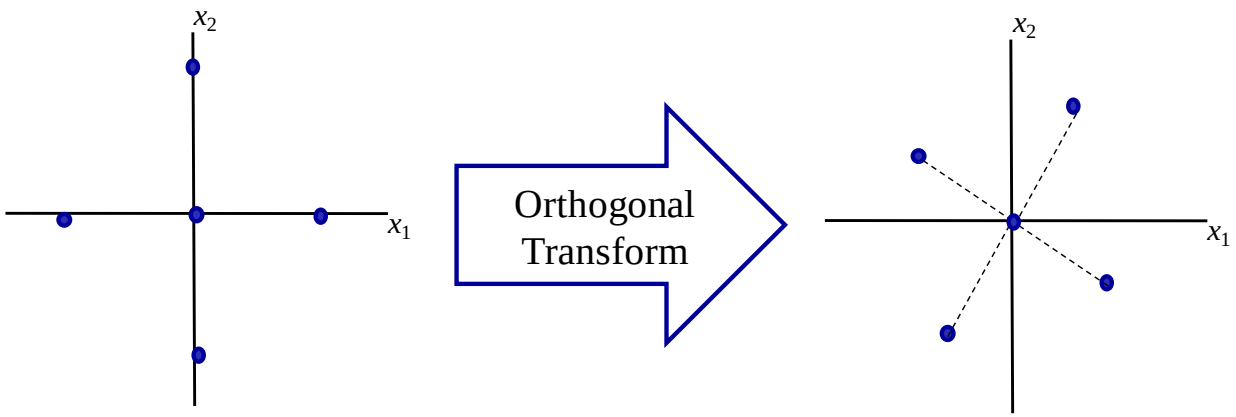

Figure 3.3. Orthogonal transform of a sigma point set.

3.4.1 Derivation of UT Accuracy

To prove the accuracy claim of the UT, we present the derivation given in [Sim06]. First, consider a random variable x described by a PDF $p(x)$. The mean of x is calculated

$$\bar{x} = \langle x \rangle = \int_{-\infty}^{\infty} x p(x) dx \quad . \tag{3.45}$$

The variance of x is

$$\sigma^2 = \int (x - \bar{x})^2 p(x) dx = \int (\Delta x)^2 p(x) dx \quad . \tag{3.46}$$

A special case occurs if x has a zero mean and symmetric PDF. A symmetric PDF implies $p(x) = p(-x)$. Uniform with zero mean and normal distributions are examples of symmetric PDFs and any uniform and Gaussian distributions, which are commonly used to describe input parameter uncertainties, can be easily transformed to uniform with zero mean and normal distribution by shifting the mean. The ith moment of x can be calculated

$$\begin{aligned} m_i = E\left[x^i\right] &= \int_{-\infty}^{\infty} x^i p(x) dx \\ &= \int_{-\infty}^{0} x^i p(x) dx + \int_{0}^{\infty} x^i p(x) dx \quad . \end{aligned} \tag{3.47}$$

Substitute $u = -x, du = -dx$ into the first term of Eq. (3.47) $u = x^i = -(-x)^i$

$$\int_{-\infty}^{0} x^i p(x)dx = -\int_{0}^{-\infty} x^i p(x)dx = \int_{0}^{\infty} (-u)^i p(-u)du \quad . \tag{3.48}$$

If i is odd, $(-u)^i = -\left(u^i\right)$ and $p(u) = p(-u)$ from symmetry of p. The result of Eq. (3.48) after quick change of variables $u = x$ is

$$\int_{0}^{\infty} (-u)^i p(-u)du = \int_{0}^{\infty} -(u^i)p(u)du = -\int_{0}^{\infty} x^i p(x)dx \quad . \tag{3.49}$$

If the result of Eq. (3.49) is substituted back into Eq. (3.47), we see that all odd moments are equal to zero if a zero mean random variable has a symmetric PDF.

Any function can be expanded as a Taylor series about a nominal linearization point. For notational simplicity, we will assume Eq. (3.40) has a single input dimension x with PDF $p(x)$ and the linearization point is the mean $\bar{x}$. The Taylor series expansion of Eq. (3.40) is

$$y = f(x) = f(\bar{x}) + \frac{\partial f(\bar{x})}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\Delta x^2 + \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\Delta x^3 + O(\Delta x^4) \quad . \tag{3.50}$$

The expectation value of Eq. (3.50) gives the mean $\bar{y}$ of y.

$$\bar{y} = \int f(x)p(x)dx = E\left[f(\bar{x}) + \frac{\partial f(\bar{x})}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\Delta x^2 + \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\Delta x^3 + O(\Delta x^4)\right]$$

$$= f(\bar{x}) + E\left[\frac{\partial f(\bar{x})}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\Delta x^2 + \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\Delta x^3 + O(\Delta x^4)\right]$$

$$= f(\bar{x}) + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\sigma^2 + O(\Delta x^4) \quad . \tag{3.51}$$

The result shown in Eq. (3.51) is obtained noting

$$E\left[\frac{\partial f(\bar{x})}{\partial x}\Delta x\right] = \frac{\partial f(\bar{x})}{\partial x}\int_{-\infty}^{\infty}(x-\bar{x})p(x)dx = \frac{\partial f(\bar{x})}{\partial x}\left(\left[\int_{-\infty}^{\infty}xp(x)dx\right]-\bar{x}\right) = 0 \quad \text{and} \tag{3.52}$$

$$E\left[\frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\Delta x^3\right] = \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\int_{-\infty}^{\infty}\Delta x^3 p(x)dx = O(\Delta x^4)\ . \tag{3.53}$$

Applying the UT to the same function $f(x)$ and setting $k = 0$, a two data point sigma set is generated

$$\{x^{(1)},x^{(2)},W^{(1)},W^{(2)}\} = \left\{\bar{x}+\sigma,\bar{x}-\sigma,\frac{1}{2},\frac{1}{2}\right\}\ , \tag{3.54}$$

and two function evaluations are performed

$$\{y^{(1)},y^{(2)}\} = \left\{f\left(x^{(1)}\right),f\left(x^{(2)}\right)\right\} = \{f(\bar{x}+\sigma),f(\bar{x}-\sigma)\}\ . \tag{3.55}$$

The UT mean $\bar{y}_u$ is calculated from Eq. (3.44) as

$$\bar{y}_u = \frac{1}{2}[f(\bar{x}+\sigma)+f(\bar{x}-\sigma)]\quad . \tag{3.56}$$

Performing a Taylor series expansion of Eq. (3.56) about $x = \bar{x}$

$$\bar{y}_u = \frac{1}{2}\left[\begin{array}{c} f(\bar{x})+f(\bar{x})+\frac{\partial f(\bar{x})}{\partial x}\sigma-\frac{\partial f(\bar{x})}{\partial x}\sigma+\frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\sigma^2+\frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\sigma^2+\frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\sigma^3- \\ \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\sigma^3+\frac{1}{4!}\frac{\partial f^4(\bar{x})}{\partial x^4}\sigma^4+\frac{1}{4!}\frac{\partial f^4(\bar{x})}{\partial x^4}\sigma^4\ldots \end{array}\right]$$

$$= f(\bar{x})+\frac{1}{2}\frac{\partial f^2(\bar{x})}{\partial x^2}\sigma^2+\frac{1}{4!}\frac{\partial f^4(\bar{x})}{\partial x^4}\sigma^4+\cdots\quad . \tag{3.57}$$

All of the odd moment terms canceled in Eq. (3.57) because the sigma point set was symmetric $\{x^{(1)} = -x^{(2)}\}$. Equation (3.57), the UT mean compares exactly with Eq. (3.51), the true mean, up to the 3rd order term regardless of the choice of sigma point set and type of input PDFs. Error starts to creep into the 4th order term.

If the input PDF is symmetric, all odd moments in Eq. (3.51) evaluate to zero and the true mean is

$$\bar{y} = f(\bar{x}) + E\left[\frac{\partial f(\bar{x})}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\Delta x^2 + \frac{1}{3!}\frac{\partial f^3(\bar{x})}{\partial x^3}\Delta x^3 + \frac{1}{4!}\frac{\partial f^4(\bar{x})}{\partial x^4}\Delta x^4 + O(\Delta x^5)\right]$$

$$= f(\bar{x}) + \frac{1}{2!}\frac{\partial f^2(\bar{x})}{\partial x^2}\sigma^2 + \frac{1}{4!}\frac{\partial f^4(\bar{x})}{\partial x^4}E[\Delta x^4] + E[O(\Delta x^6)] \quad . \tag{3.58}$$

Comparing Eq. (3.58) to the UT mean of Eq. (3.57), the error in the 4th order term is introduced because $E[\Delta x^4] \neq \sigma^4$. However, we can calculate $E[\Delta x^4]$ for any PDF. For example, is *x* is a normal distribution, the 4th moment is equal to $3\sigma^4$. Through careful selection of *k* in Eq. (3.43), a particular sigma point set and associated weighting factors can be chosen to minimize the error in the 4th order and higher terms given the type of input parameter PDF. A similar derivation can show the UT variance approximation is accurate up to the 3rd order term of the true variance.

The UT generates sigma points around the mean input so the variation of the inputs from the linearization point of the expansion, which is usually the mean input, could be small. For nonlinear functions that are sufficiently smooth, the higher order derivatives should be small. Furthermore, the higher order terms in the Taylor series expansion are divided by increasingly larger factorials. All of these factors could minimize the error introduced into the UT from the higher order terms, thus improving the overall accuracy of the UT.

3.4.2 UT and Nonlinear System Dynamics

The UT presented in the previous section is applicable to propagating distributions through any general nonlinear function *f*(). Next we consider the case where *f*() is a function describing nonlinear system dynamics which was the original application of the UT and will be relevant to the development of a dynamic code surrogate in Chapter 5. Dynamic systems are often modeled as discrete time systems where the system state *x* is advanced over discrete time steps from the current system state estimate

$$x(k+1) = f[x(k), \boldsymbol{u}(k), v(k), k] \quad , \tag{3.59}$$

where $\boldsymbol{u}$ is the input parameter vector, v is the model noise or model uncertainty and k is the current time step. From an initial condition $x(0)$ at time zero, the model uncertainty introduces uncertainty into the subsequent state estimates. Thus, the uncertainty of $x(k)$ and the model uncertainty $v(k)$ must be propagated through $f()$ resulting in a distribution for $x(k+1)$. The UT provides an efficient means to accurately estimate the mean and variance of the new state estimate $x(k+1)$.

3.5 Code Surrogates in BEPU Methodologies

In 1988, the U.S. Nuclear Regulatory Commission (NRC) approved the revised rule on the acceptance of Emergency Core Cooling System (ECCS) which offered the option to use the evaluation model prescribed in Appendix K of 10CFR50.46 or a best-estimate (BE) computer code for safety analysis of NPPs provided that the uncertainty of the BE results are quantified. The evaluation model required the use of conservative models and assumptions that would sometimes lead to overly conservative results resulting in the performance of as designed safety systems not meeting the regulations. BE codes using realistic physical models were attractive tools for NPP owners in the licensing process and power uprate applications. However, applying BE computer codes to analyze full scale NPP transients and rigorously assessing the results was a new and challenging regulatory and engineering endeavor.

The NRC and its contractors developed the code scaling, applicability, and uncertainty (CSAU) methodology and applied it to a LBLOCA in a Westinghouse four-loop pressurized water reactor (PWR) demonstrating the use of BE codes for ECCS analyses [NRC89]. The CSAU methodology implemented a three element process in 14 steps. The first element, Requirements and Code Capabilities, identifies the TH phenomena associated with selected transient and NPP type and compares with the code capabilities and possible limitations. A phenomena identification and ranking table (PIRT) is generated through a systematic approach that evaluates the effects and importance of phenomena on primary safety criteria. In the second element, Assessment and Ranging of Parameters, the code capabilities to calculate the important process and phenomena are assessed by simulating integral effects experiments and separate

effects experiments with the code. Scale-up capability, nodalization, bias, and range of parameter uncertainties are determined. In the third element, Sensitivity and Uncertainty Analyses, bias and uncertainty identified in second element are combined with uncertainties of the initial NPP state to obtain a statement of total uncertainty in the safety criteria for the transient.

The result of the first two elements in the CSAU methodology is a set of uncertain input parameters defined by PDFs that must be propagated through the BE code simulation to see the variation in the output system parameters relevant to the safety criteria. Two general approaches for the input uncertainty propagation exist. The first is input parameter sampling and direct computer code simulation which yields a data set from which statistical properties of the output distribution can be estimated. The second approach is to develop a code surrogate and generate the output distribution from Monte Carlo simulation using the surrogate. In the CSAU demonstration, response surfaces were constructed from 184 PCT values obtained from eight TRAC simulations and output PDFs of the PCT were generated by sampling the response surfaces through Monte Carlo methods. For the time period, performing eight TH code calculations of a full scale NPP transient model required the state-of-the-art in computational resources. The response surface models were linear regression models with up to 4th degree polynomial basis functions. The response surfaces were limited to the seven most important input parameters. Westinghouse developed a Best Estimate Plus Uncertainty (BEPU) methodology similar to the CSAU implementing response surfaces of low order polynomial linear regression models and applied the methodology to the analysis of a LBLOCA for the AP600 NPP design [You98, Zha98].

Advances in computer science and technology through the widespread availability of the personal computer and increasingly faster processors since the first demonstration of the CSAU methodology allows for greater flexibility in uncertainty propagation. Consequently, reactor vendors and nuclear industry research organizations have developed BEPU methodologies generally following the CSAU methodology but use the direct simulation method with nonparametric order statistics to estimate probability levels of the output parameter distributions instead of response surface methods [Mar05,Fre08,Gla08]. The 95%/95% limits, the 95th percentile at a 95% confidence level, are calculated from Wilks' formula [Wil41]. A 95% confidence level generally assures a conservative estimate of the 95th percentile and qualifies the 95%/95% limits as the statement of total uncertainty. Wilks' formula allows all uncertainty

contributors to be randomly sampled from their underlying PDFs for each code run and only require a total of 59, 93, or 124 simulations for 1st, 2nd, and 3rd order Wilks' formula, respectively. While Wilks' formula is simple to apply and requires finite computational resources making it attractive to use in BEPU methodologies, the numerical limits obtained are subject to statistical fluctuations inherent with any randomly sampled data set of limited size and the full output parameter PDFs are not calculated.

3.5.1 LBLOCA Application of GPM and ACE

In this section, we summarize the uncertainty analysis of a LBLOCA for a PWR presented in [Fyn12, Fyn13]. The uncertainty analysis demonstrates the relevance of the GPM and ACE algorithm to nuclear reactor safety calculations and BEPU methodologies while demonstrating the process of surrogate construction. The MARS code [KAE09] was selected as the system TH code to model a 200% double-ended guillotine break in a cold leg of an OPR1000, a two-loop PWR. The PCT during the blowdown phase of the LBLOCA was the safety parameter of interest in the study. The GPM and ACE algorithm were used to generate response surfaces for the blowdown PCT as a function of 20 input parameters to the MARS model of the OPR1000. Table 3.1 lists the 20 input parameters and associated PDFs used in the study. The parameters cover physical models in the MARS code and plant parameters important to a fuel behavior and TH conditions in the core during the LBLOCA blowdown phase. A training data set with 400 data points was generated by sampling the input parameter distributions and performing MARS simulations. A 200 sample LHS design was selected for coverage of the input parameter space supplemented by an additional 200 random samples. A cross-validation test set of 111 random samples was also generated for a total 511 MARS simulations performed. The MOSAIQUE uncertainty analysis software [Lim11] was used to automate the LHS and random sampling schemes, input file generation, and execution of the MARS simulations.

Table 3.1. MARS Input Parameters and PDFs.

Code Input x_i	Independent parameter	PDF	Uncertainty range (min/max or σ)	Reference Value	Units
1	Gap Width	Uniform	0.05-.134	0.092	mm
2	Cladding Roughness	Normal	+/- 0.3	0.5	micron
3	Pellet Roughness	Normal	+/- 0.5	2	micron
4	Fuel Thermal Conductivity	Normal	+/-10%	Table	%
5	Cladding Thermal Conductivity	Normal	+/-12%	Table	%
6	Fuel Heat Capacity	Normal	+/-10%	Table	%
7	Cladding Heat Capacity	Uniform	90 - 110 %	Table	%
8	Rod Internal Pressure	Uniform	2.0-15.0	8	MPa
9	Power Peaking Factor	Uniform	1.5054-1.7147	1.61	
10	Operating Plant Power	Normal	+/-2%	2815	MWt
11	Pump 2-1 phase head multiplier	Uniform	0-1.0	RELAP5 Default	
12	Pump Torque Multiplier	Uniform	0-1.0	Table	
13	Pump Inlet K-factor	Uniform	0.02-0.8	0.41	
14	Pump Momentum of Inertia	Uniform	4487.2-5484.4	4985.8	kg/m^2
15	CCFL Model	Uniform	0-1.0	0 (Wallis)	
16	Chen's Nucleate Boiling HT Multiplier	Normal	+/-11.6%	1	
17	AECL lookup CHF Multiplier	Normal	+/-37%	1	
18	Transition Boiling Multiplier	Normal	+/-16%	1	
19	Film Boiling HT Multiplier	Normal	+/-18%	1	
20	Decay Heat	Normal	+/- 6.6%	ANS79-1 , 1.02 multiplier	

The ACE response surface is the one-dimensional transformations for each of the twenty input parameters and dependent variable, the blowdown PCT, derived from the 400 data point training set. Figure 3.4 shows the ACE transformations. A GPM was constructed using the squared exponential covariance function given in Eq. (3.25). The Gaussian Processes for Machine Learning (GPML) Toolbox [Ras10], an open source GPM code package written in Matlab and GNU Octave programming languages, was used to learn the covariance function hyperparameter values. To improve the numerical stability and interpretability of the GPM, all of the input parameters were normalized by scaling linearly from -0.5 to 0.5 corresponding to the min/max for uniformly distributed parameters or +/-3σ for normally distributed parameters. The PCT values were shifted by the sample mean so that the GPM has a zero mean function. Table 3.2 lists the characteristic length scale hyperparameters r_i for each input parameter dimension learned by the ARD minimization algorithm in the GPML software. The GPM response surface is the mean function given by Eq. (3.8). The inversion of the 400 × 400 covariance matrix was only performed once and multiplied by the 400 × 1 vector of PCT values to yield the 400 × 1 vector of smoothed PCT values which is stored. To make a prediction, the 1 × 400 prediction covariance vector must be generated and product taken with the smoothed PCT vector. Each element of the prediction covariance requires the vector and matrix operations involving the 20 dimension input parameter space.

Table 3.2. GPM and ACE model parameters corresponding to MARS inputs and sensitivity rankings.

Code Input x_i	Independent parameter	GPM Length Scale Hyperparameter	Range of ACE Transformation $\Delta\phi(x_i)$	ACE Sensitivity Rank
1	Gap Width	2.711	1.645	**2**
2	Cladding Roughness	5.146	0.649	**10**
3	Pellet Roughness	1.674	1.335	**4**
4	Fuel Thermal Conductivity	0.657	3.108	**1**
5	Cladding Thermal Conductivity	1.543	0.674	**9**
6	Fuel Heat Capacity	2.949	0.415	**12**
7	Cladding Heat Capacity	8.714	0.421	**11**
8	Rod Internal Pressure	58.603	0.097	**17**
9	Power Peaking Factor	2.428	0.713	**8**
10	Operating Plant Power	3.505	1.182	**5**
11	Pump 2-1 phase head multiplier	0.869	0.794	**7**
12	Pump Torque Multiplier	5.049	0.066	**18**
13	Pump Inlet K-factor	6.937	0.038	**19**
14	Pump Momentum of Inertia	403.611	0.032	**20**
15	CCFL Model	2.239	0.158	**15**
16	Chen's Nucleate Boiling HT Multiplier	14.556	0.104	**16**
17	AECL lookup CHF Multiplier	1.529	1.152	**6**
18	Transition Boiling Multiplier	5.048	0.303	**13**
19	Film Boiling HT Multiplier	2.713	1.575	**3**
20	Decay Heat	16.777	0.243	**14**

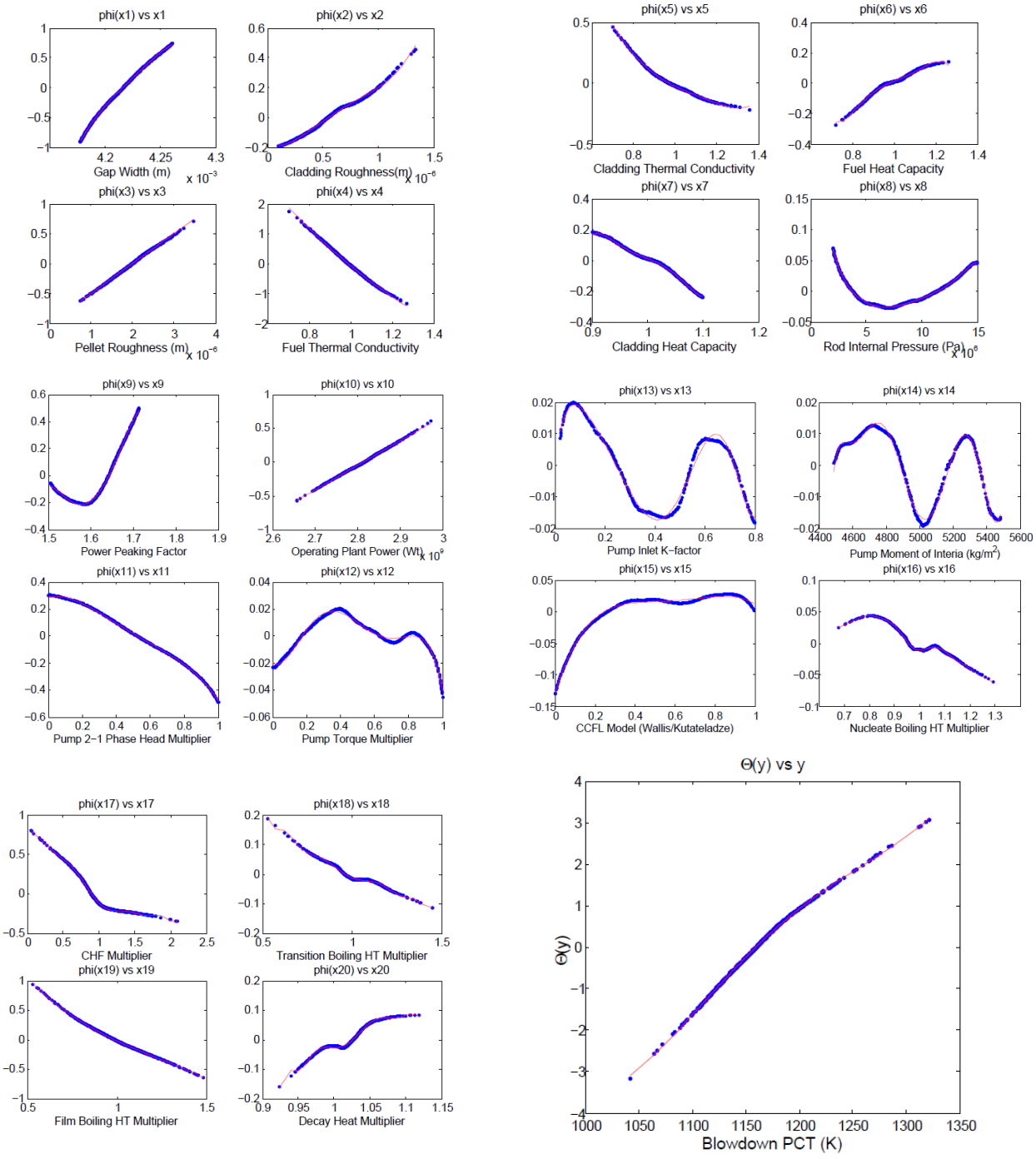

Figure 3.4. ACE transformations for LBLOCA blowdown PCT response surface.

A practical advantage of the GPM with a squared exponential covariance function and the ACE algorithm one-dimensional transformations over conventional regression models is that sensitivity information is automatically obtained during surrogate construction. The inverse of the characteristic length scales, which have been conveniently normalized, can be interpreted as sensitivity coefficients for the input dimensions. The ACE transformations are all normalized to the single dimension transform space. The absolute range of each $\phi_i(x_i)$ is a measure of what fraction of the range of y, through $\theta(y)$, is explained by the variation in x_i. This is a function both the sensitivity of y to x_i and the assumed uncertainty range of x_i so the range of the transforms are not sensitivity coefficients from a strict definition sense but do offer quantitative sensitivity or importance information. A range of 1 in the transforms space corresponds to approximately a range of 45 K for PCT. Table 3.2 lists the ACE transformation range for each input and the associated sensitivity ranking. A semi-quantitative comparison of the GPM length scales and ACE sensitivity rankings reveal that both methods are generally identifying the same parameters as sensitive or insensitive. For example, the fuel thermal conductivity has the shortest GPM length scale and is ranked most sensitive by the range of the ACE transforms whereas the pump moment of inertia has the longest GPM length scale and is rank last by the ACE transform. An inspection of the least sensitive input transformations in Figure 3.4 show highly nonlinear or oscillatory behavior but over a small range of ϕ indicating these variables are free parameters that are adjusted arbitrarily by the ACE algorithm to account for any residual unexplained variance.

Figure 3.5 compares the predictive accuracy of the GPM and ACE response surfaces to the MARS simulation cross-validation test set. The surrogate predictions are tightly scattered about the 45 degree goodness of fit line. The maximum observed deviation of the GPM and ACE response surfaces from the test set PCT was 27 K and 25 K, respectively. The cross-validation data provides evidence that the surrogates are accurate and unbiased and when considered coupled with the agreement of the sensitivity information, we can confidently move forward with applying the surrogates to the UQ of the blowdown PCT.

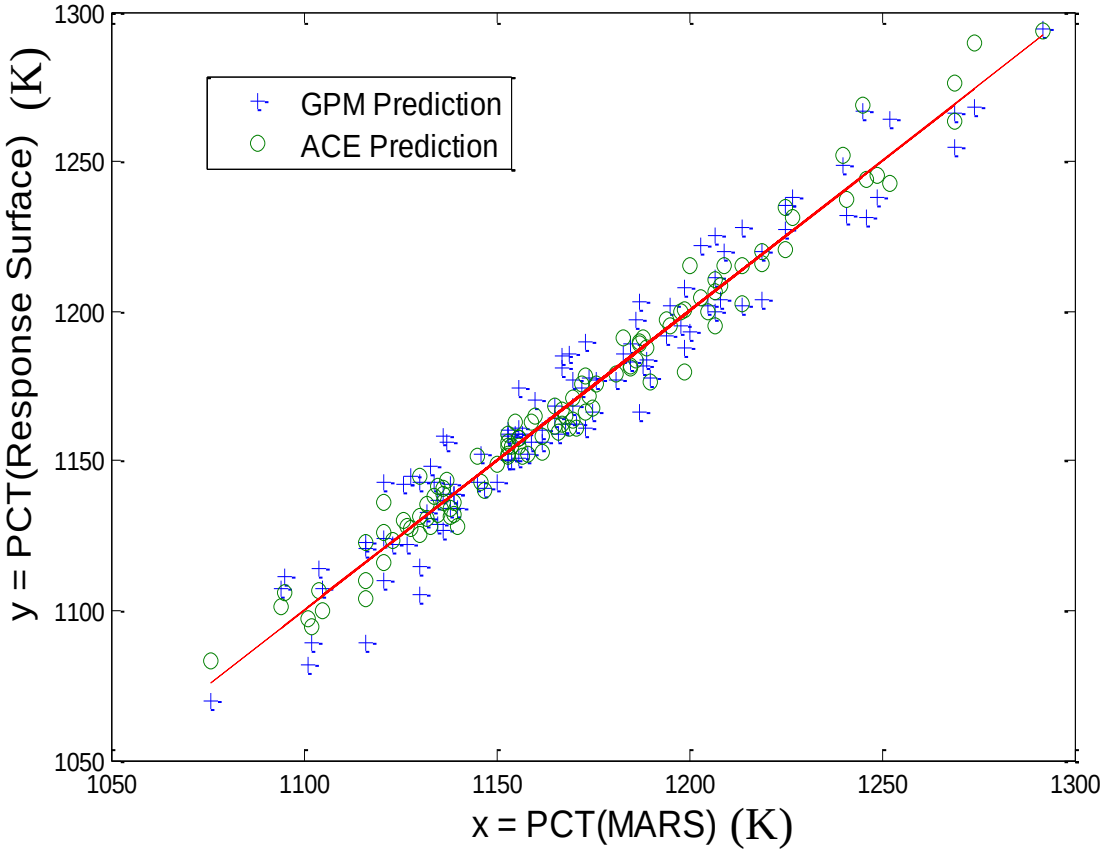

Figure 3.5. Cross-validation of GPM and ACE response surface PCT predictions to MARS code results.

The uncertainty of the blowdown PCT can be quantified by obtaining a PDF of PCT. The input parameter distributions are sampled through a Monte Carlo method and the PCT values are obtained at very low cost from the response surface models. Figure 3.6 shows the PCT PDFs from 100,000 random samples with PCT calculated with the GPM and the ACE models. From the GPM derived PDF, the mean value for PCT of the LBLOCA is 1161.9 K and the 95th percentile is 1235 K. From the ACE derived PDF, the mean value for PCT of the LBLOCA is 1162.7 K and the 95th percentile is 1236 K. The decimal place used in the mean PCT values is included for comparison purposes only and does not quantify the precision of the surrogate predictions. Despite being obtained from two different response surface models, the statistical data obtained from the PCT PDFs are in very close agreement. The mean PCT value obtained from the response surface analysis is approximately 1162 K which is significantly greater than the PCT value of 1072 K from the reference case MARS simulation. Many of the

input parameter uncertainties used in this study are uniform distributions so the nominal values chosen for these parameters may bias the result if only a reference case is considered. Thus performing an uncertainty analysis to determine the true mean and other statistical information about the output parameters is an important element of BE modeling and simulation.

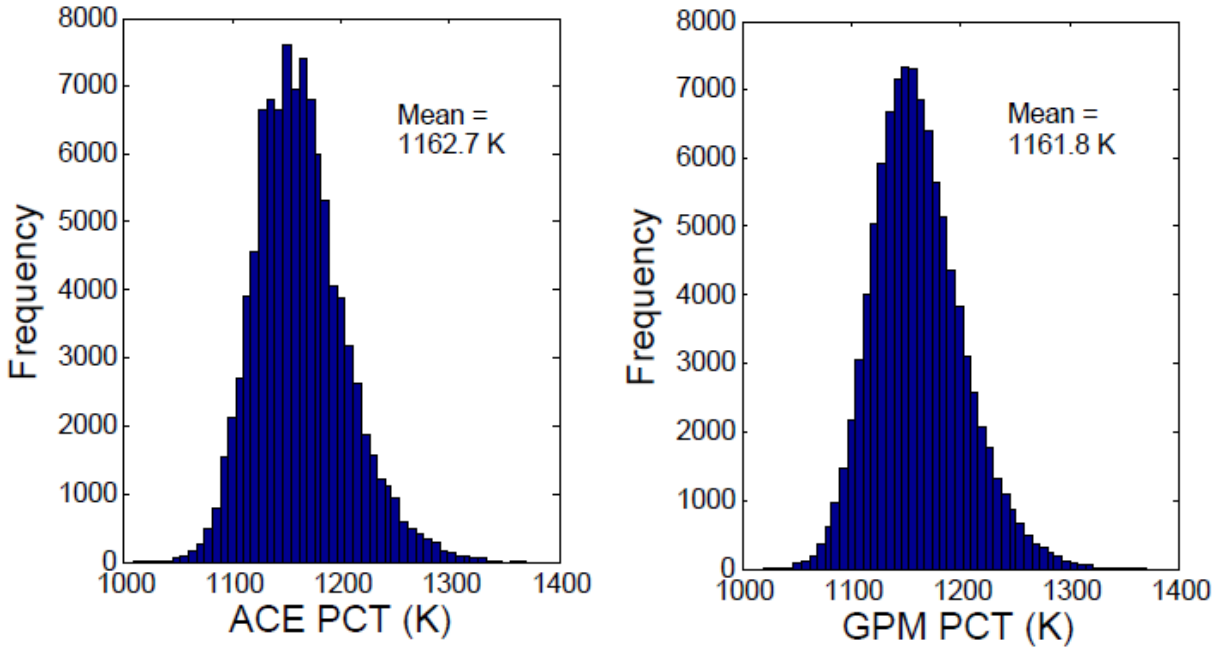

Figure 3.6. PDFs of blowdown PCT from 100,000 random samples evaluated with ACE and GPM response surfaces.

3.5.2 LBLOCA Application of UT

Once a surrogate has been constructed, it is a computationally efficient tool that can be used for a variety of purposes including UQ. We now use the ACE response surface presented in Section 3.5.1 as a benchmarking tool to assess the applicability of the UT as a deterministic sampling based approach for UQ. Specifically, we apply the UT to the estimate the mean and variance of LBLOCA blowdown PCT and perform a parametric study to determine the optimal set of sigma points through the selection of parameter k in Eqs. (3.42) and (3.43) and the use of orthogonal transforms. In place of the MARS code, all simulations are performed with the ACE response surface so the UT results can be directly compared with the ACE derived PCT PDF.

The UT transform should be able to accurately estimate the mean and variance of the PCT requiring only 41 samples and evaluations of the ACE response surface. The central point $\boldsymbol{x}^{(0)}$ is equal to the mean values $\bar{\boldsymbol{x}}$ of the input parameter distributions given in Table 3.1 and the covariance matrix $\boldsymbol{P}$ is a 20 × 20 diagonal with the diagonal entries equal to the variance of the input PDFs. To test the sensitivity of the UT to the specific chosen sigma point set, seven different matrix square roots were selected by multiplying the unique square root $\boldsymbol{A}_u$ by a 20 × 20 orthogonal matrix. Table 3.3 summarizes the orthogonal matrices used to define the matrix square roots. For matrix square roots $\boldsymbol{A}_2$ through $\boldsymbol{A}_4$, specific transforms such as the Hartley transform and discrete cosine transform are used to produce symmetric orthogonal matrices. For matrix square roots $\boldsymbol{A}_5$ through $\boldsymbol{A}_7$, orthogonal matrices were randomly generated by performing singular value decomposition (SVD) on randomly generated 20 × 20 matrices.

Table 3.3. Orthogonal transforms used for sigma point set generation.

$\boldsymbol{A}_i = \boldsymbol{A}_u \boldsymbol{Q}$	$\boldsymbol{Q}$
$\boldsymbol{A}_1$	$\boldsymbol{I}$
$\boldsymbol{A}_2$	$Q(i,j)=\sqrt{(2/(n+1))}\sin((i*j*\pi)/(n+1))$
$\boldsymbol{A}_3$	$Q(i,j)=\sin(2\pi(i-1)(j-1)/n)+\cos(2\pi(i-1)(j-1)/n)$
$\boldsymbol{A}_4$	$Q(i,j)=\sqrt{(2/n)}\cos((i-1/2)(j-1/2)\pi/n)$
$\boldsymbol{A}_5$, $\boldsymbol{A}_6$, $\boldsymbol{A}_7$	SVD of random 20 × 20 matrix

For each orthogonal transform, the mean and variance of the PCT were calculated with the UT for varying values of k. Figure 3.7 compares the UT mean estimates to the true mean of 1163 K from the ACE response surface benchmark as a function of k. Figure 3.8 compares the UT variance estimates to the true variance of 1702 K^2 corresponding to a standard deviation of 41.25 K as a function of k. The orthogonal transform $\boldsymbol{A}_1$ which is equal to the unique square root $\boldsymbol{A}_u$ performed poorly for all k. For the symmetric orthogonal transforms, the sigma point sets using $k = 2$ and $k = 0$ gave the most accurate mean and variance estimates, respectively. The symmetric orthogonal transforms still performed well for the range $k = [-4,6]$. For the randomly generated orthogonal transforms, the sigma point sets using $k = -4$ gave the most accurate mean and variance estimates. The randomly generated orthogonal transforms still performed well for the range $k = [-6,4]$.

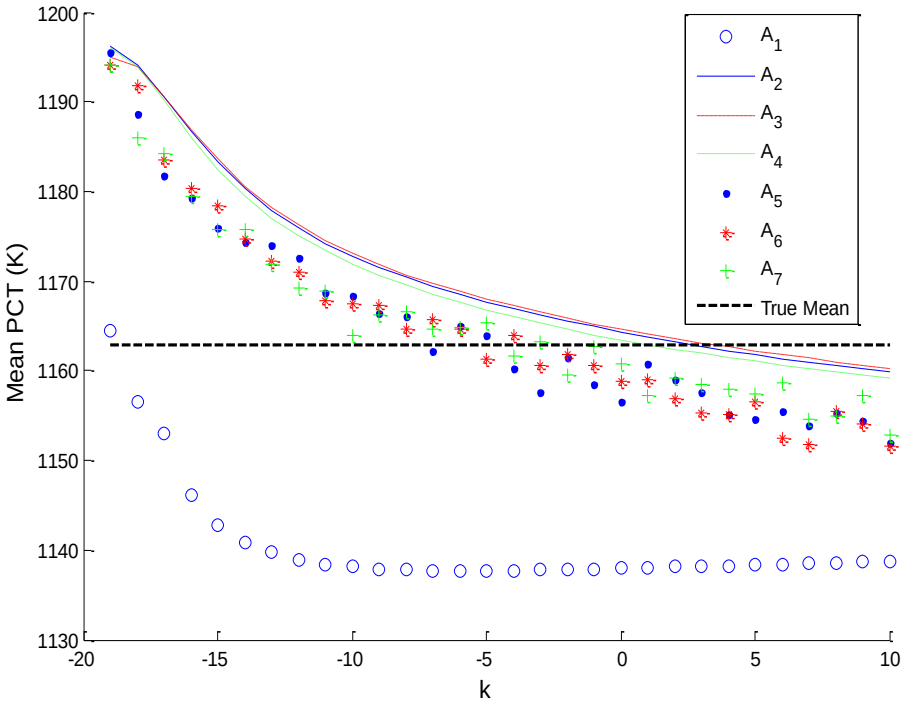

Figure 3.7. Accuracy of UT mean estimate for different matrix square roots and *k*.

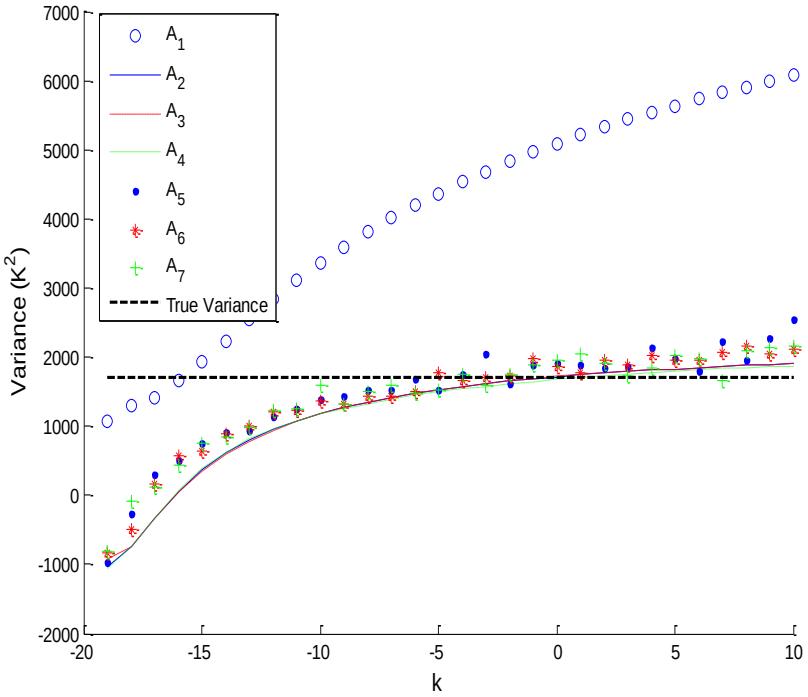

Figure 3.8. Accuracy of UT variance estimate for different matrix square roots and *k*.

The UT appears to be robust and performs well for sigma sets derived from matrix square roots that are transformed by an orthogonal matrix and a range of values of k. For the LBLOCA example, values of k approximately equal to zero gave accurate estimates of the mean and variance of the PCT. Given the definition of $W^{(0)}$ in Eq. (4), the magnitude of the weight applied to the central point is minimized for $k = 0$. The PCT $y^{(0)}$ calculated at the central point is 1145 K which is significantly less than the true mean of 1163 K. For $k > 0$, $W^{(0)}$ is positive and the UT mean is biased less than the true mean because a positive weight has been assigned to the point $y^{(0)}$ which is an underestimation of the true mean. Conversely, for $k < 0$, a negative weight on $y^{(0)}$ biases the UT mean above the true mean. Therefore, it is not surprising that sigma point sets which minimize the contribution of the central point in the estimation of the mean and variance give the best results. However, this conclusion may only be valid for the particular nonlinear function considered in this study, the ACE response surface, and the assumed PDFs for the input parameters.

References

[Ada09] B.M Adams, W.J. Bohnhoff, K.R. Dalbey, J.P. Eddy, M.S. Eldred, D.M Gay, K. Haskell, P.D. Hough, and L.P. Swiler, "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual," Sandia Technical Report SAND2010-2183, December 2009.

[Bai99] C.A.L. Bailer-Jones, H.K.D.H. Bhadeshia, and D.J.C. MacKay, "Gaussian Process Modeling of Austenite Formation in Steel," *Materials Science and Technology*, **15**, 287(1999).

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

[Cle79] W. S. Cleveland, "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, **74**, 829(1979).

[Fre08] C. Frepoli, “An Overview of Westinghouse Realistic Large Break LOCA Evaluation Model,” *Science and Technology of Nuclear Installations*, **2008**, (2008).

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

[Fri91] J.H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, **19**, 1(1991).

[Fyn12] D.A. Fynan, K.I. Ahn, H.G. Lim and J.C. Lee, “Uncertainty Quantification of LBLOCA PCT for a Pressurized Water Reactor by ACE Algorithm and Gaussian Process Model,” *Proc. 18th Pacific Basin Nuclear Conference*, Busan, Korea, March 18-23, 2012.

[Fyn13] D.A. Fynan, K.I. Ahn, and J.C. Lee, " Unscented Transform for Approximating Mean and Covariance of Distributions in BEPU Methodologies, *Transactions of the American Nuclear Society*, **109**, (2012).

[Gla08] H. Glaeser, "GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications," *Science and Technology of Nuclear Installations*, **2008**, (2008).

[Joh90] M.E. Johnson, L.M. Moore, and D. Ylvisaker, "Minimax and Maximin Distance Designs," *Journal of Statistical Planning and Inference*, **26,** 131(1990).

[Jul04] S.J. JULIER and J.K. UHLMANN, "Unscented Filtering and Nonlinear Estimation," *Proceedings of the IEEE*, **92**(3), 401(2004).

[KAE09] KAERI, “MARS Code Manual Volume I: Code Structure, System Models, and Solution Methods,” KAERI/TR-2812/2004, Korea Atomic Energy Research Institute (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).

[Lim11] H.G. Lim, S.H. Han, and J.J. Jeong, “MOSAIQUE – A Network Based Software for Probabilistic Uncertainty Analysis of Computerized Simulation Models”, *Nuclear Engineering and Design*, **241**, 1776(2011).

[Mar05] R.P. Martin and L.D. O'Dell, "AREVA's Realistic Large Break LOCA Analysis Methodology," *Nuclear Engineering and Design*, **235**, 1713(2005).

[McK79] M.D. McKay, R.J. Beckman and W.J. Conover, "Comparison of Three Methods for Selecting of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, **21**, 23(1979).

[Nea96] R.M. Neal, R.M., *Bayesian learning for neural networks*, No. 118 in Lecture Notes in Statistics, Springer, New York, 1996.

[NRC89] USNRC, Quantifying Reactor Safety Margins: Application of CSAU to a LBLOCA, NUREG/CR-5249, USNRC, Washington, DC, USA, 1989.

[Ras06] C.E. Rasmussen and K.I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[Ras10] C.E. Rasmussen and H. Nickisch, "Gaussian Process for Machine Learning (GPML) Toolbox," *Journal of Machine Learning Research*, **11**, 3011(2010).

[Sim06] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.

[Wil41] S.S. Wilks, "Determination of Sample Sizes for Setting Tolerance Limits," *The Annals of Mathematical Statistics*, **12**, 91(1941).

[You98] M.Y. Young, S.M. Bajorek, M.E. Nissley and L.E. Hochreiter, "Application of Code Scaling and Uncertainty Method to the Large Break Loss of Coolant", *Nuclear Engineering and Design*, **186**, 39(1998).

[Zan01] J.R. Van Zandt, "A More Robust Unscented Transform," *Proceedings of SPIE 2001*, **4473**, 371(2001).

[Zha98] J. Zhang, S.M. Bajorek, R.M. Kemper, M.E. Nissley, N. Petkov and L.E. Hochreiter, "Application of the WCOBRA/TRAC Best-Estimate Methodology to the AP600 Large-Break LOCA Analysis," *Nuclear Engineering and Design*, **186**, 279(1998).

CHAPTER IV

RELAP5 MODEL OF ULCHIN 3&4 NUCLEAR POWER PLANT

This chapter will introduce the modeling of a LBLOCA for a pressurized water reactor (PWR). The optimized power reactor 1000 MWe (OPR1000), formally known as the Korean Standard Nuclear Power Plant (KSNP), is the selected reference plant design for the analysis of a LBLOCA. The Ulchin 3&4 NPP (UCN3&4) is the specific reference plant we are analyzing. RELAP5 is the system TH code used to simulate the LBLOCA. In section 4.1, we will discuss the design and safety features of UCN3&4. An overview of RELAP5 will be given in section 4.2. The RELAP5 model of UCN3&4 will be presenting in section 4.3 and the TH behavior of the plant during the LBLOCA will be discussed.

4.1 Description of Ulchin 3&4 Nuclear Power Plant

In 1971, construction began for the Kori-1 NPP marking the start of commercial nuclear power in the Republic of Korea. Kori-1 is a two-loop, Westinghouse PWR and was constructed under a turnkey contract with foreign companies providing the majority of the engineering design, component manufacturing, and construction. The following NPP projects in Korea including several three-loop Westinghouse PWRs and four CANDU-6 heavy water reactors at Wolsong Nuclear Power Complex involved Korean firms taking over some of the primary project management and construction roles and component manufacturing. In the late 1980s, construction began for Yonggwang Units 3&4 (YGN3&4), 1000 MWe PWRs based on the Combustion Engineering (CE) two-loop System 80 design. These reactors were constructed under the third phase of Korea's nuclear power program which called for technology self-reliance in the Korean nuclear industry. YNG3&4 included a technology transfer agreement between CE and the Korea Electric Power Corporation (KEPCO). YNG3&4 became the reference plant for

the first Korean standardized NPP design, the OPR1000. The first OPR1000s are UCN3&4 which began operation in 1998 and 1999, respectively. Subsequently, an additional eight OPR1000 units are operating or nearing completion in Korea.

The OPR1000 is a two-loop PWR with a rated power of 2815 MWt corresponding to 1000 MWe [KEP96]. Each loop consists of a steam generator (SG), two cold legs, one hot leg, and two reactor coolant pumps (RCP). The reactor core is normally loaded with 177 fuel assemblies with a 16 × 16 design. The reactor operates on a 12 to 18 month refueling cycle. Figure 4.1 shows the configuration of the reactor coolant system (RCS) of the OPR1000.

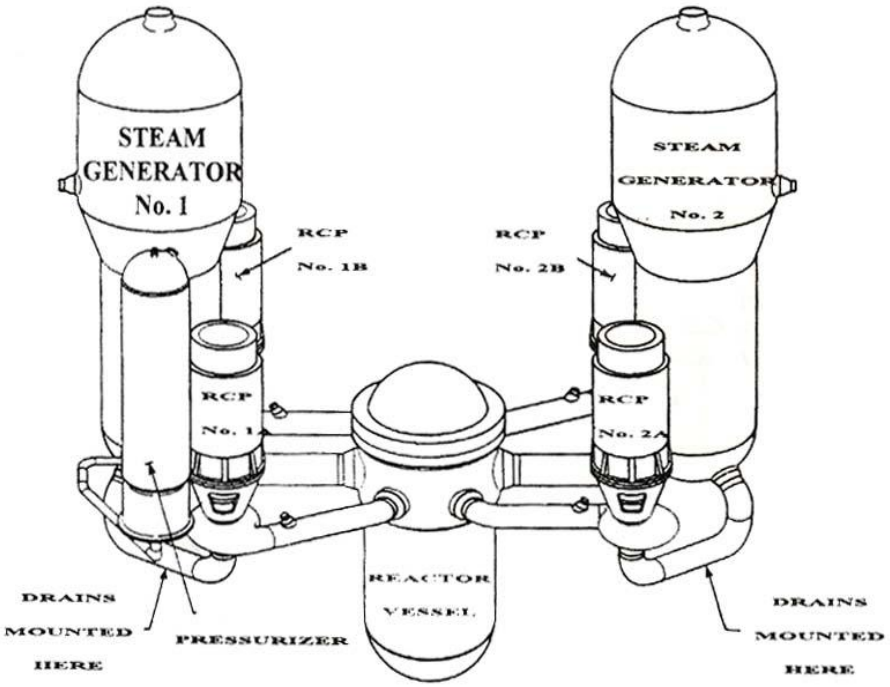

Figure 4.1. Reactor coolant system of the OPR1000. From [KEP96].

4.1.1 Engineered Safety Systems of the OPR1000

The OPR1000 has several engineered safety systems to prevent the release of radiation to the environment during an accident. This section will discuss three systems important to the loss-of-coolant accidents (LOCA), the safety injection system (SIS), containment spray system (CSS), and auxiliary feedwater system (AWFS). The SIS is responsible for injecting borated

water into the reactor coolant system (RCS) to flood and cool the core following a LOCA to prevent cladding and fuel damage. The SIS also provides long term heat removal from the core. The major components of the SIS are four safety injection tanks (SITs), high-pressure safety injection (HPSI) system and a low-pressure safety injection (LPSI) system.

The SITs are designed to rapidly inject large volumes of water into the RCS immediately following a LBLOCA to reflood the core. Each SIT contains approximately 14,000 gallons of borated water and is pressurized with nitrogen gas. During the blowdown of a LBLOCA, most of the core water inventory flashes to steam and RCS rapidly depressurizes as large amounts of water and steam flows out the break. When the RCS pressure drops below the SIT pressure setpoints, the SITs to inject into the cold legs. The SITs are the only source of SI water until other active components requiring AC power in the SIS can start.

The HPSI system consists of two HPSI pumps and associated valves and piping. The pumps are AC powered, horizontal, centrifugal type with a design flow of 815 gpm. The HPSI pumps are designed to match the boil off rate at the earliest time of recirculation during a LBLOCA. The shutoff head is 1775 psig so the pumps can deliver SI water to the RCS during small-break LOCAs where the RCS pressure may remain high. The HPSI pumps are designed to automatically start after receiving the Safety Injection Actuation Signal (SIAS) and pump SI water from the refueling water storage tank (RWST) to each of the four discharge legs of the cold leg piping through SI lines. After the RWST depletes and the Recirculation Actuation Signal (RAS) is received, the HPSI pump suctions are automatically realigned to the containment sumps. Eventually, the operators can manually align the HPSI system to the hot legs for simultaneous cold and hot leg injection.

The LPSI system consists of two LPSI pumps and associated valves and piping. The pumps are AC powered, vertical, single stage, centrifugal type with a design flow of 4200 gpm at a design head of 335 ft. The LPSI pumps are also used in the shutdown cooling system (SCS) and are sized for SCS functional requirements. The LSPI pumps automatically start after receiving the SIAS and provide large mass flow rates of SI water to the RCS during the injection phase of a LBLOCA.

The CSS provides cooling spays of water to the containment atmosphere to reduce containment pressure by condensing steam and scrubs the atmosphere of volatile fission products. The CSS is designed to limit the containment pressure and temperature during the

blowdown during a LOCA or a steam line break inside the containment. The CSS consists of two redundant trains each with a CS pump, shutdown cooling heat exchanger (SCX), spray headers and associated valves and piping. The CS pumps are AC powered, vertical, single stage. centrifugal type with a design flow of 3890 gpm. The CSS automatically starts upon receiving the Containment Spray Actuation Signal (CSAS) on high containment pressure. During the injection phase, the water source for the CSS is the RWST. During recirculation, the CS pumps draws water from the containment sumps and passes the water through the SCX before the spray headers. This mode of operation allows the removal heat from the containment to the ultimate heat sink.

The AFWS supplies feedwater to the SGs whenever the main feedwater system is unavailable and reactor cooldown needs to be performed. During normal shutdown or accident conditions, heat removal through the SGs can cool down the primary system so the AFWS is designed to provide enough feedwater to cool down the plant to conditions where the SCS can be utilized. The AFWS consists of two trains, one for each SG. Each train has a turbine-driven (TD) pump and a motor-driven (MD) pump to pump feedwater from the condensate storage tank (CST) to the SG. During a LBLOCA, the AFWS is of secondary importance to the SIS and CSS but is very important during other operational transients and accidents.

4.2 Overview of RELAP5

RELAP5 (Reactor Excursion and Loss of Coolant Analysis Program) [NRC01] is a system TH code designed for modeling single and two-phase flow in light water reactors for a variety of transient conditions including design basis accidents and operational transients. RELAP5 is used in the nuclear industry for licensing calculations, PSA, evaluating accident mitigation strategies and operator procedures, and TH experiment design and analysis. Calculations can be performed in standalone mode or with RELAP5 coupled to subchannel, neutronics, or containment codes.

The RELAP5 code structure is modular with three top level blocks consisting of input, transient/steady-state, and strip functions. The input block processes and checks input data from user supplied input cards or from a RELAP5 restart file. The strip block simply extracts simulation results from the restart file and prints the tabular data to a file that can be used in

postprocessing. The actual simulation is performed by the transient block which solves a system of coupled differential equations representing reactor kinetics and TH behavior of the system.

The physical NPP structures and components, the reactor vessel, piping, pumps, valves, fuel rods, SG tubes, etc., are modeled as hydrodynamic volumes and heat structures. Water, steam and air reside in and flow to and from the hydrodynamic volumes that are interconnected by hydrodynamic junctions. The heat structures represent the solid boundaries of the hydrodynamic volumes. Heat structures can be heat sources, heat sinks, or permit heat transfer from one hydrodynamic volume to another as is the case with heat transfer through SG tube walls from the tube side to the shell side of the SG.

4.2.1 RELAP5 Hydrodynamic Model

The RELAP5 hydrodynamic model is a one-dimensional, transient, two-fluid, nonequilibrium, nonhomogeneous model and is executed by the transient block to solve for the two-phase flow conditions in the hydrodynamic volumes. The model is derived from six basic field equations, two phasic continuity equations, two phasic momentum equations, and two phasic energy equations. With the addition of two mass continuity equations to include noncondensable gases and dissolved boron, the six basic field equations are solved to obtain eight dependent variables describing time-averaged and volume-averaged two-phase flow. The eight dependent variables are the pressure (P), phasic specific internal energies (U_g, U_f), void fraction (α_g), phasic velocities (v_g, v_f), noncondensable quality (X_n), and the boron density (ρ_b). The subscripts g and f denote the vapor and liquid phases, respectively.

The phasic continuity equations for conservation of mass are

$$\frac{\partial}{\partial t}\left(\alpha_g \rho_g\right) + \frac{1}{A}\frac{\partial}{\partial x}\left(\alpha_g \rho_g v_g A\right) = \Gamma_g \quad \text{and} \tag{4.1}$$

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f\right) + \frac{1}{A}\frac{\partial}{\partial x}\left(\alpha_f \rho_f v_f A\right) = \Gamma_f \quad , \tag{4.2}$$

with volume fraction (α), density (ρ), flow area (A), and mass generation rate (Γ). The mass generation term Γ allowing for the transfer of mass between the vapor and liquid phases

representing vapor generation or condensation. In the absence of any mass sources or sinks, the liquid generation term is equal to the negative of the vapor generation $\Gamma_f = -\Gamma_g$.

The momentum equations are

$$\alpha_g\rho_g A\frac{\partial v_g}{\partial t}+\frac{1}{2}\alpha_g\rho_g A\frac{\partial v_g^2}{\partial x}=-\alpha_g A\frac{\partial P}{\partial x}+\alpha_g\rho_g B_x A-(\alpha_g\rho_g A)FWG(v_g)+\Gamma_g A(v_{gi}-v_g)$$
$$-(\alpha_g\rho_g A)FIG(v_g-v_f)-C\alpha_g\alpha_f\rho_m A\left(\frac{\partial(v_g-v_f)}{\partial t}+v_f\frac{\partial v_g}{\partial x}-v_g\frac{\partial v_f}{\partial x}\right) \quad (4.3)$$

and

$$\alpha_f\rho_f A\frac{\partial v_f}{\partial t}+\frac{1}{2}\alpha_f\rho_f A\frac{\partial v_f^2}{\partial x}=-\alpha_f A\frac{\partial P}{\partial x}+\alpha_f\rho_f B_x A-(\alpha_f\rho_f A)FWF(v_f)+\Gamma_f A(v_{fi}-v_f)$$
$$-(\alpha_f\rho_f A)FIF(v_f-v_g)-C\alpha_f\alpha_g\rho_m A\left(\frac{\partial(v_f-v_g)}{\partial t}+v_g\frac{\partial v_f}{\partial x}-v_f\frac{\partial v_g}{\partial x}\right), \quad (4.4)$$

The subscript *i* and *w* denotes the vapor/liquid interface and wall, respectively. The force terms are aggregated on the right hand side of Eqs. (4.3) and (4.4) and are the pressure gradient, body force, wall friction, momentum transfer at interface from mass transfer, interface frictional drag, and virtual mass force, respectively. The force terms are parameterized by the body force (B_x), the wall frictional drag coefficients (*FWG*, *FWF*), the interface drag coefficients (*FIG*, *FIF*), and the virtual mass coefficient (*C*) which is flow regime dependent. Equations (4.3) and (4.4) were developed under the assumption that momentum effects in the fluid flow are secondary to mass and energy conservation in reactor safety analysis so a less exact formulation for the momentum equations could be used [NRC01].

The energy conservation equations are

$$\frac{\partial}{\partial t}(\alpha_g\rho_g U_g)+\frac{1}{A}\frac{\partial}{\partial x}(\alpha_g\rho_g U_g v_g A)=-P\frac{\partial\alpha_g}{\partial t}-\frac{P}{A}\frac{\partial}{\partial x}(\alpha_g v_g A)+$$
$$Q_{wg}+Q_{ig}+\Gamma_{ig}h_g^*+\Gamma_w h_g'+DISS_g \quad (4.5)$$

and

$$\frac{\partial}{\partial t}(\alpha_f\rho_f U_f)+\frac{1}{A}\frac{\partial}{\partial x}(\alpha_f\rho_f U_f v_f A)=-P\frac{\partial\alpha_f}{\partial t}-\frac{P}{A}\frac{\partial}{\partial x}(\alpha_f v_f A)+$$
$$Q_{wf}+Q_{if}-\Gamma_{ig}h_f^*-\Gamma_w h_f'+DISS_f \quad . \quad (4.6)$$

Recognizing the heat transfer and flow conditions can be different in the bulk fluid than in the boundary layer near the wall of a heat structure, Eqs. (4.5) and (4.6) separate Γ_g into two components, the vapor generation rate at the vapor/liquid interface in the bulk fluid (Γ_{ig}) and the vapor generation near the wall (Γ_w). The phasic enthalpies (h) associated with each vapor generation term use the superscripts (*) and (') to designate interface and wall, respectively. Similarly, the heat transfer rates (Q) are separated into the wall transfer rates for each phase and the interface terms. The phasic energy dissipation terms (DISS) are the summation of the wall friction and pump effects.

4.2.2 RELAP5 Heat Conduction Model

The heat conduction model in RELAP5 calculates the temperature profiles in the heat structures and the heat transfer rates to the fluid in the hydrodynamic volumes. The temperature profiles $T(\boldsymbol{x},t)$ are obtained from the integral form of the heat conduction equation

$$\iiint c_p(T,\boldsymbol{x})\frac{\partial T}{\partial t}(\boldsymbol{x},t)dV = \iint k(T,\boldsymbol{x})\boldsymbol{\nabla}T(\boldsymbol{x},\mathrm{t})\cdot \mathrm{d}\mathbf{s} + \iiint S(\boldsymbol{x},t)dV \quad , \tag{4.7}$$

with volumetric heat capacity (c_p), thermal conductivity (k), and internal heat source (S). The boundary conditions

$$-k\frac{\partial T}{\partial n} = h(T - T_{sk}) \qquad \text{and} \tag{4.8}$$

$$\frac{\partial T}{\partial n} = 0 \tag{4.9}$$

are applied to Eq. (4.7) where n is defined as the outward unit normal vector from the heat structure surface. Equation (4.8) represents heat transfer out of the surface as a function of the surface temperature (T), the sink temperature (T_{sk}), and the heat transfer coefficient (h). Equation (4.9) is the symmetric or insulated boundary condition. For heat transfer to an hydrodynamic volume, h is calculated by the heat transfer correlation package which contains empirical correlations for convective, nucleate boiling, transition boiling and film boiling heat

transfer. Alternatively, time-dependent heat transfer rates or heat transfer coefficients can be directly input as table data.

Equation (4.7) with Eq. (4.8) is solved by finite differences in one-dimensional rectangular, cylindrical, or spherical geometry. For rectangular geometry where the heat structure is assumed to be a rectangular solid, a calculation mesh is applied in the positive x direction. For cylindrical and spherical geometry where the heat structure is a cylindrical annulus or sphere, a radial mesh is applied. Mesh points are usually placed at the desired intervals, the external boundaries of the heat structure, and material interfaces within the structure. Over each mesh interval, the material type and $S(\boldsymbol{x})$ are supplied by the user and are assumed constant. RELAP5 calculates $c_p(T)$ and $k(T)$ for each mesh interval using an average temperature calculated from the bounding mesh point temperatures and temperature dependent tabular data for c_p and k for each material type.

4.2.3 RELAP5 Point Kinetics Model

The point kinetics model is used in RELAP5 to calculate the time-dependent fission power and the decay power of the reactor. The main assumption of the point kinetics approximation is the neutron flux which is proportional to power can be separated into space and time functions. Point kinetics are adequate for transients where the space distribution of the flux remains relatively constant and the change in the magnitude of the reactor power is reflected in the time function of the flux. The point kinetic equations are a set of coupled differential equations for the time-dependent neutron flux (φ) and the concentration of delayed neutron precursors groups (C_i)

$$\frac{d}{dt}\varphi(t) = \frac{[\rho(t)-\beta]}{\Lambda}\varphi(t) + \sum_{i=1}^{6}\lambda_i C_i(t) + S \qquad \text{and} \qquad (4.10)$$

$$\frac{d}{dt}C_i(t) = \frac{\beta f_i}{\Lambda}\varphi(t) - \lambda_i C_i(t) \ , \ i = 1,2,\ldots,6 \ , \qquad (4.11)$$

Data based on the core design and fuel cycle must be supplied to define the kinetics parameters reactivity (ρ), effective delayed neutron fraction (β), prompt neutron generation time (Λ), decay

constant (λ), neutron source (S), and fraction of delayed neutrons in group i (f_i). Six delayed neutron groups are commonly used for LWR analysis. The reactor power (P) from fission can be calculated from the neutron flux, energy released per fission (Q), and the macroscopic fission cross section (Σ_f) by

$$P(t) = Q\Sigma_f \varphi(t) \quad . \tag{4.12}$$

The time-dependent reactivity $\rho(t)$ is dependent on many system variables so a feedback model is used to calculate ρ as the reactor system evolves through a transient. The feedback model is assumed to be separable and individual reactivity contributions from the coolant density, coolant temperature, and fuel temperature or doppler reactivity are summed from tabular data. The active core usually spans many hydrodynamic volumes and heat structures so axial temperature and density gradients exist. Equations (4.10) and (4.11) are derived in zero-dimensional space so the reactivity contributions from the system variables that vary axially and radially for the case of the fuel temperature in core region must be lumped into single terms. Volume and heat structure weighting factors are used to obtain weighted point values of the reactivity contributions based on the neutronic importance of each core region. Time-dependent reactivity data representing a SCRAM or control rod ejection can also be incorporated into the feedback model.

The decay power from fission products and actinides is calculated from a model based on the American Nuclear Society Proposed Standard ANS 5.1, Decay Energy Release Rates Following Shutdown of Uranium-Fueled Thermal Reactors, revised October 1973, or the American National Standard for Decay Heat Power in Light Water Reactors, ANSI/ANS-5.1-1979. Both models use the power history from the kinetics model and the user supplied fuel cycle history. A fission product yield factor multiplier to adjust the decay power for conservative or best estimate problems.

The solution of the point kinetics equations gives the immediate fission power of the reactor through Eq. (4.12). The total power is obtained from the sum of the fission power and decay power. The total power becomes a multiplier variable to the heat conduction model for the internal heat source in the heat structures representing the fuel pins. The internal heat

sources in the fuel pin structures are defined to reflect the spatial power distribution of the core so the total power scales the internal heat sources.

4.3 RELAP5 Model of Ulchin 3&4 Nuclear Power Plant

The RELAP5 model for the UCN3&4 was developed at the Korea Atomic Energy Research Institute (KAERI) from the small-break LOCA RELAP5 model for YGN3&4 [Jeo93]. The UCN3&4 model can simulate a LBLOCA in either loop of the RCS. The break can occur in the hot leg or the discharge leg of the cold leg between the RCP and the reactor vessel inlet.

4.3.1 Reactor Coolant System Nodalization

The RCS is modeled by 250 hydrodynamic volumes, 280 hydrodynamic junctions, and 259 heat structures. Figure 4.2 is the nodalization diagram for UCN3&4 RELAP5 model. The reactor vessel is comprised as series of connected volumes representing the inlet annulus, downcomer, lower plenum, active core, core bypass, outlet plenum and upper head region. The reactor internal structures and vessel walls are modeled as heat structures. The active core is modeled as a single channel by a vertical pipe hydrodynamic volume with 14 axial segments coupled to a single cylindrical heat structure representing the fuel assemblies. To preserve the three-dimensional flow and heat transfer characteristics of the core, the cross-sectional flow area of the pipe equals to interstitial space between all of the fuel pins and the surface area of the heat structure equals the total surface area of all of the fuel pins. The heat structure has the same radial dimensions of a single fuel pin reflecting the clad thickness, gap width and fuel pellet radius. The axial power distribution in the fuel is set to the top-skewed cosine shape specified in the Final Safety Evaluation Report (FSAR) for LBLOCA analysis [KEP96]. The pin power is scaled by the total reactor power. The axial and radial temperature profile obtained for the heat structure represents the profile from a single fuel pin with the core-averaged power density. The hottest fuel pin in the core is also modeled as a separate cylindrical heat structure.

The SGs are modeled by hydrodynamic volumes representing the primary side and the shell side of the SG separated by a cylindrical heat structure representing the SG tube walls. Similar to the modeling of the fuel assemblies in the core volume, the SG tube bundle is modeled

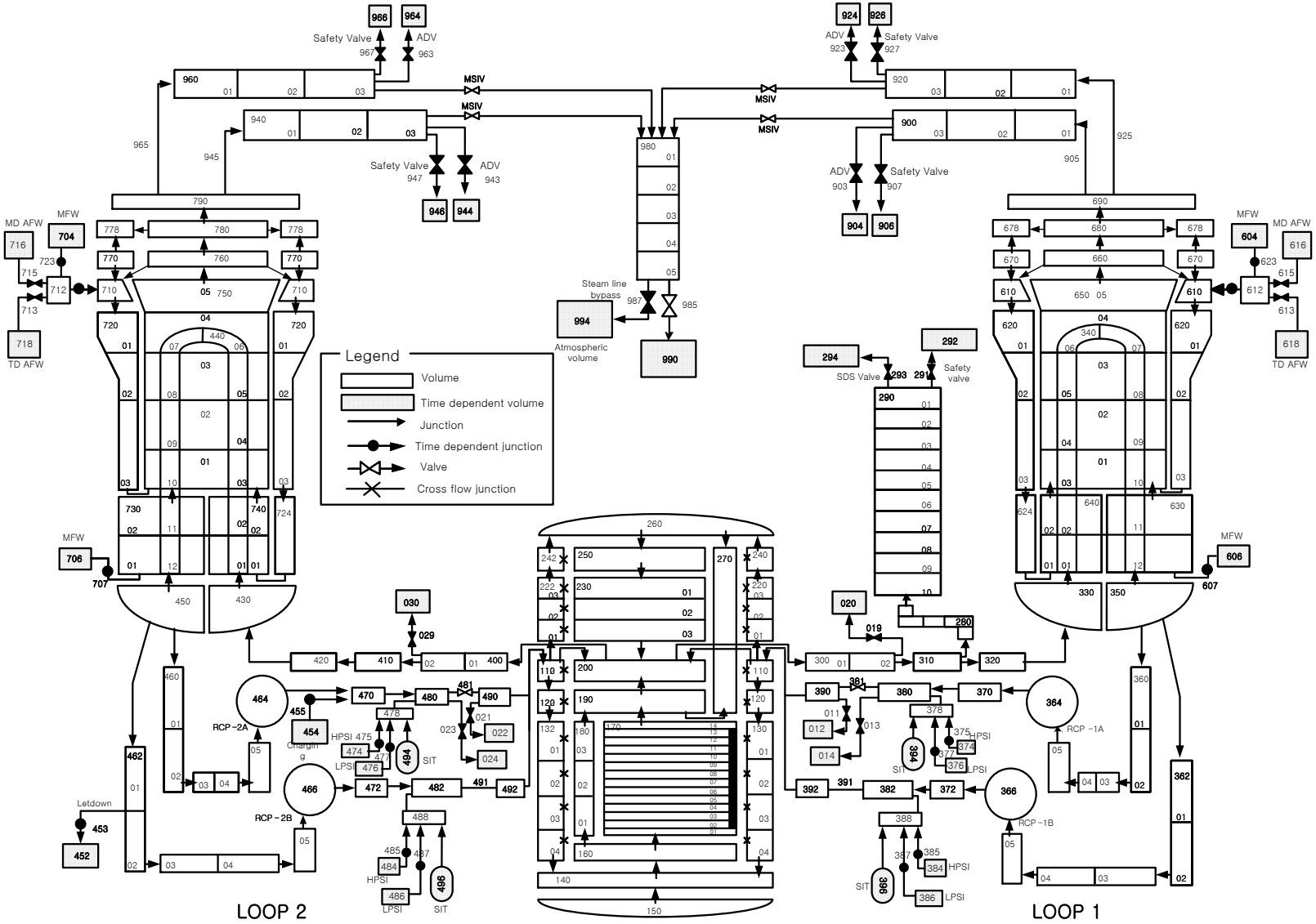

Figure 4.2. Nodalization diagram for UCN 3&4.

as a single pipe hydrodynamic volume with the cross-sectional flow area and heat transfer area of all of the individual tubes conserved. The pipe has vertical and angled segments to preserve the U-shape of the tube bundle. Heat is transfer from the tube bundle volume through tube wall heat structure to the shell side volumes.

For some NPP components with specific functions, RELAP5 provides individual component models. These component models require additional user input and are treated with special numerical techniques during the simulation compared to the other generalized hydrodynamic volumes. The RCPs are modeled with the pump component model. The pump component model interacts with the hydrodynamic model by adding a pump head term to the mixture momentum equations and a dissipation term to the energy equations that is a function of the pump torque and angular momentum. Empirical homologous head and torque curves and two-phase degradation curves are used by the pump model to calculate the pump performance.

4.3.2 Safety System Modeling

The active safety systems of UCN3&4, mainly the AFWS, HPSI system, and LPSI system, are modeled as time-dependent volumes and junctions which are best described as time-dependent boundary conditions to the physical plant model. A time-dependent volume allows the user to define a hydrodynamic volume containing fluid with specified properties such as temperature and pressure as a function of time. The mass flow rate through a time-dependent junction is specified by the user as a function of time or any other system variable such as the pressure in the connecting volume. In the UCN3&4 model, the RWST and containment sump are modeled as time-dependent volumes and the HPSI and LPSI pumps are modeled as time-dependent junctions connecting the RWST or containment sump volumes to the cold leg (CL) piping. The RWST and containment sump volumes are essentially infinite water sources that define the temperature of the water flowing through the junctions. Flow rates curves for the HPSI and LPSI pumps are input as functions of the pressure in the discharge legs where the SI lines connect to the CL piping. The AFWS is modeled in the same way with the condensate storage tank (CST) modeled as time-dependent volumes and the MD pumps and TD pumps as time-dependent junctions connected to the shell side of the SGs. The SITs are modeled with the accumulator component model.

RELAP5 is usually not used for containment analysis so the containment volumes and CSS are not explicitly modeled. Since the CSS draws water from the RWST during the injection phase of a LBLOCA, the flow rate of the CSS pumps is tracked along with the flow of the HPSI and LPSI pumps in order to calculate the water level in the RWST. The water level is used to determine the RAS time and transition from the injection phase to the recirculation phase.

In the UCN3&4 plant, automatic instrumentation and control (I&C) signals or manual operator actions can actuate or stop safety systems, valves, pumps and other active components. Passive components such as safety relief valves automatically open and close when system thresholds are reached. The trip system in RELAP5 is extensively used to model I&C signals and the actuation or stopping of components in the UCN3&4 model. The trip system allows the evaluation of logical statements to produce a true (+1) or false (-1) response. Within component models, trip status (true or false) determines the component state. For example, to control a time-dependent junction or valve, a trip in a true status opens the junction or valve while a false status closes.

The trip logical statements can be variable or logical trip. Variable trips evaluate numerical statements comparing system or control variables. Logical trips compare the true false results of other variable or logical trips. Table 4.1 summarizes the variable and logical trips that produce the SIAS signal in the LBLOCA. Trip 496 controls all of the time-dependent junctions of the HPSI and LPSI systems connecting to the CLs. Physically, trip 496 is the 30 second delay time to start the HPSI and LPSI pumps after the SIAS signal is generated from the low pressure in the pressurizer (trip 427). Other signals and system actuation such as the RAS and AFWS starting are generated in a similar manner to the SIAS logic.

Table 4.1. Trip table for Safety Injection Actuation Signal.

426	**p**	**290100000**	**lt**	**null**	**0**	**121.5e5**	**n**	**1.**	*** PZR LO PRES R 121.5psi**
427	**p**	**290100000**	**lt**	**null**	**0**	**121.0e5**	**n**	1.	* PZR LO PRES S 121.0psi
626	**426**	**and**	**627**			**n**		1.	* PZR LO PRES
627	**427**	**or**	**626**			**n**		1.	* "PZR LO PRES TRIP"
531	**time**	**0**	**gt**	**null**	**0**	**1.0e6**	**l**	1.	* "Manual reactor trip"
740	**531**	**or**	**627**			**l**		1.	* "Rx Trip"
532	**time**	**0**	**gt**	**timeof**	**740**	**2.0**	**l**	1.	* "Rods begins to drop"
496	**time**	**0**	**gt**	**timeof**	**427**	**30.0**	**l**	1.	* "SI On with 30s Delay"

4.3.3 Break Modeling

In order to model a break in the CL or HL piping, the hydrodynamic junction connecting the two volumes of the piping at the break location is replaced with a valve component and two additional valves connecting each volume to time-dependent volumes representing the containment compartment. During the steady-state calculation and before the break is initiated, the valve connecting the two volumes remains open representing an intact pipe and the other valves representing the break area are closed. At the time of the break, the valve at the junction is closed and the two break valves are opened creating new flow paths from the piping volumes to the containment volumes. The time-dependent containment volumes are sinks for the water and steam that exit the break. The break flow is not explicitly coupled to the containment conditions but the pressure of the containment volumes obtained from a separate containment calculation are input as time-dependent data to provide feedback to the break flow.

References

[Jeo93] J. J. Jeong and K. S. Ha, "Calculation Note: RELAP5/MOD3 Input Data Generation of Yonggwang Units 3 & 4 for SBLOCA Analysis," Korea Atomic Energy Research Institute, Advanced Reactor System Development Department, June 1993.

[KEP96] KEPCO, "Ulchin Units 3&4 Final Safety Analysis Report," Korea Electric Power Corporation, Final Safety Analysis Report, 1996.

[NRC01] USNRC, "RELAP5/MOD3.3 Code Manual," NUREG/CR-5535/Rev.1, U.S. Nuclear Regulatory Commission, 2001.

CHAPTER V

DYNAMIC SURROGATE MODEL FOR RECIRCULATION PHASE LBLOCA

The GPM and ACE PCT surrogate models presented in Chapter 3 are examples of response surfaces that are static input-output nonlinear mappings. Although the PCT was calculated from the dynamic simulation of the LBLOCA with the MARS code, the surrogates predict the variation in the PCT directly from variations in the input parameters without having to compute the complete time-dependent system behavior. Obtaining only the limiting safety parameters distributions is acceptable for licensing calculations where the final result is to determine if regulatory safety limits are met. Conversely, in dynamic PSA applications, the system state evolution under various different operation conditions coupled to interactions with component transitions and human operator actions that can lead to safe or unsafe plant states is the desired result. Obtaining the system state evolution trajectories requires that complete simulations are performed. A dynamic code surrogate that can predict time-dependent system behavior would significantly reduce the computational burden compared to direct calculations with a TH code.

5.1 General Framework for Dynamic Surrogate Model Development

Dynamic models used in scientific and engineering applications are most commonly systems of differential equations that define the instantaneous rates of change of system state variables with respect to input variables and the current estimate of the system state variables. The RELAP5 UCN3&4 model can be simply described as a very large system of several thousand coupled differential equations, the field equations for every hydrodynamic volume and the heat conduction equation for the heat structures. The system of equations must be solved by

numerical methods such as finite differencing using time-marching methods to calculate subsequent system states from states at previous time steps. Forward Euler, backwards Euler, and Runge-Kutta are common time-marching methods. Numerical methods generally involve the linearization of the system equations with time step size constrained by stability and truncation error considerations. A single dynamic model can often produce very different system state trajectories depending on the time-dependent boundary conditions and initial conditions selected as input parameters. Figure 5.1 shows a general example of time-dependent input parameter and corresponding system state trajectories.

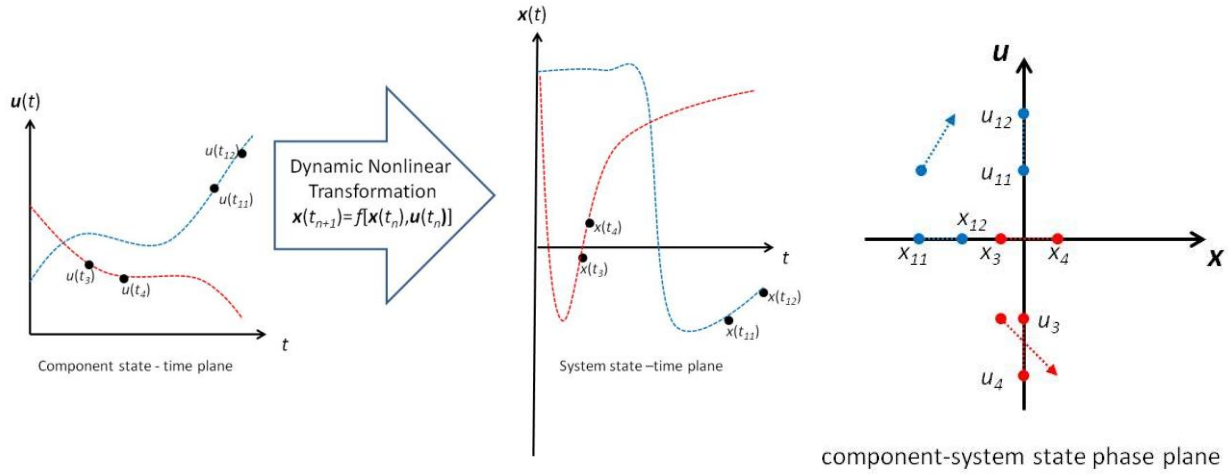

Figure 5.1. Illustration of input parameter trajectories $u(t)$ and system state trajectories $x(t)$ propagated through a dynamic model and the component state and system state nonlinear phase plane.

Surrogate development always requires training on data obtained from experiment or a BE simulation model. Through the training process correlations between the data are learned and the functional relationships are assigned to the input and output parameters. For a general dynamic model, the data would be unique time-dependent input parameter trajectories and the corresponding system state trajectories illustrated in Figure 5.1. Training a surrogate in the time domain would be difficult noting at a time t, several trajectories can have the same $\boldsymbol{u}(t)$ but different $\boldsymbol{x}(t)$ or the same $\boldsymbol{x}(t)$ but different $\boldsymbol{u}(t)$ resulting in training on a multivalued function. An alternative approach is to consider training the surrogate in the nonlinear phase space of the

component state represented by $\boldsymbol{u}(t)$ and the system state shown in Figure 5.1. The nonlinear phase space treats time implicitly so the functional relationships between changes in $\boldsymbol{u}$ to changes in x may be easier to resolve.

5.1.1 Lumped Parameter Models for NPP Analysis

For some NPP transients, reactor physics and TH behavior can be accurately described with simple mathematical models. In these cases, a computationally expensive coupled neutronics and TH code calculation can be replaced by a lumped parameter model. Lumped parameters models usually represent the reactor kinetics with the point kinetics equations coupled to TH feedback calculated from average fuel temperatures and coolant densities. The lumped parameter model manifests as a relatively simple system of differential equations that can be solved analytically or numerically with finite differencing to obtain reactor power, temperatures, and coolant properties as a function of time. A recent study [Mar06] has incorporated effective parameters of into a lumped parameter model that are calibrated by fitting to BE code simulated data.

Lumped parameters models could be viewed as a special case of a dynamic code surrogate replacing coupled neutronics and TH code models. In this case, lumped parameter models are analogous to the linear regression models discussed in Chapter 3 where if *a priori* information about the underlying function, i.e. physics of reactor kinetics and feedback, a simple mathematical model represented by the set of differential equations can perform well. As with linear regression models, if the lumped parameter model is extrapolated to a transient where it cannot represent the underlying physics, the model will break down and give poor results.

We propose a general framework for developing dynamic code surrogates for NPP applications. Borrowing from the discussion of dynamic models and lumped parameter models, the dynamic surrogate takes the form of a discrete time dynamic system given in Eq. (3.59) with input parameters representing known time-dependent boundary and initial conditions. The functional form of the system equations are learned from data from BE code simulations using the ACE algorithm. By training on BE code data, properties of the high-fidelity solution is retained in the surrogate in contrast to the lumped parameter model where the physics of the

problem are simplified at the very beginning of the model development process with the derivation of the model equations and simplified representation of the NPP.

The framework consists of simplifying the plant nodalization to a few key control volumes representing regions of pertinent TH phenomena. After identifying all sources and sinks of mass and energy representing boundary conditions, conservation of mass and energy equations can be set up for each control volume from which independent variables can be identified to form the basis for surrogate equations. The ACE algorithm is applied to data sets from the BE code simulations of the transient with variations in the boundary conditions of the model. The ACE transformations become the dynamic surrogate model. In the next section, we implement the general framework to develop a dynamic surrogate to predict the TH behavior in the core during the recirculation phase of a HL-LBLOCA for UCN3&4 for a spectrum of degraded HPSI system states.

5.2 Dynamic Surrogate for Water Level During Recirculation Phase HL-LBLOCA

The PCT studied in Chapter 3 is one regulatory limit that must be analyzed for LBLOCAs. The long term cooling of the core must also be verified by calculating the core temperature after successful initial operation of ECCS. The core temperature must be maintained at an acceptably low value and decay heat removed. For UCN3&4, the HPSI and LPSI systems and supporting subsystems discussed in Chapter 4 must operate successfully in several operational configurations including injection mode, recirculation mode, and simultaneous hot and cold leg injection for successful long term cooling.

During the injection phase, the HPSI and LPSI systems in their initial configuration supply large volumes of cold water from the RWST to the RCS resulting in single phase flow in the RCS and subcooling in the core. During the recirculation phase, the LPSI pumps are tripped and the water source for the HPSI system is switched to the containment sump where water and condensed steam from the break and containment spray water collects. Recirculation can be initiated as early as 20 minutes after a LBLOCA occurs so the relatively high decay power in the core, reduced SI flow rate after trip of LPSI pumps, higher temperature of the sump water compared to the RWST, and reflux heat transfer conditions in the intact SG loop transition the TH conditions in the RCS to two phase flow and steam production in the core.

5.2.1 Macroscopic Balance Statement for ACE Input Parameter Selection

For the recirculation phase of the HL-LBLOCA, the UCN3&4 RCS is represented by three volumes, the reactor core, the cold leg piping and downcomer, and the SG in the intact loop. Figure 5.2 shows the simplified RCS nodalization. The core can be further subdivided into two regions, a subcooled region with volume *V* where the flow is single phase and a two phase region. In the intact SG loop, saturated steam generated from the core can either be condensed from heat transfer to the shell side of the SG or superheated from the reflux conditions that exist early on during recirculation when the shell side temperature has not cooled down below the primary side saturation temperature. Counter current flow in the hot leg, loop seals and subsequent loop seal clearing develop in the intact loop. Safety injection water from the containment sump is delivered by the HPSI pumps to the cold leg piping where the water mixes with any residual water and steam in the piping and any water flow that has circulated through the intact SG loop.

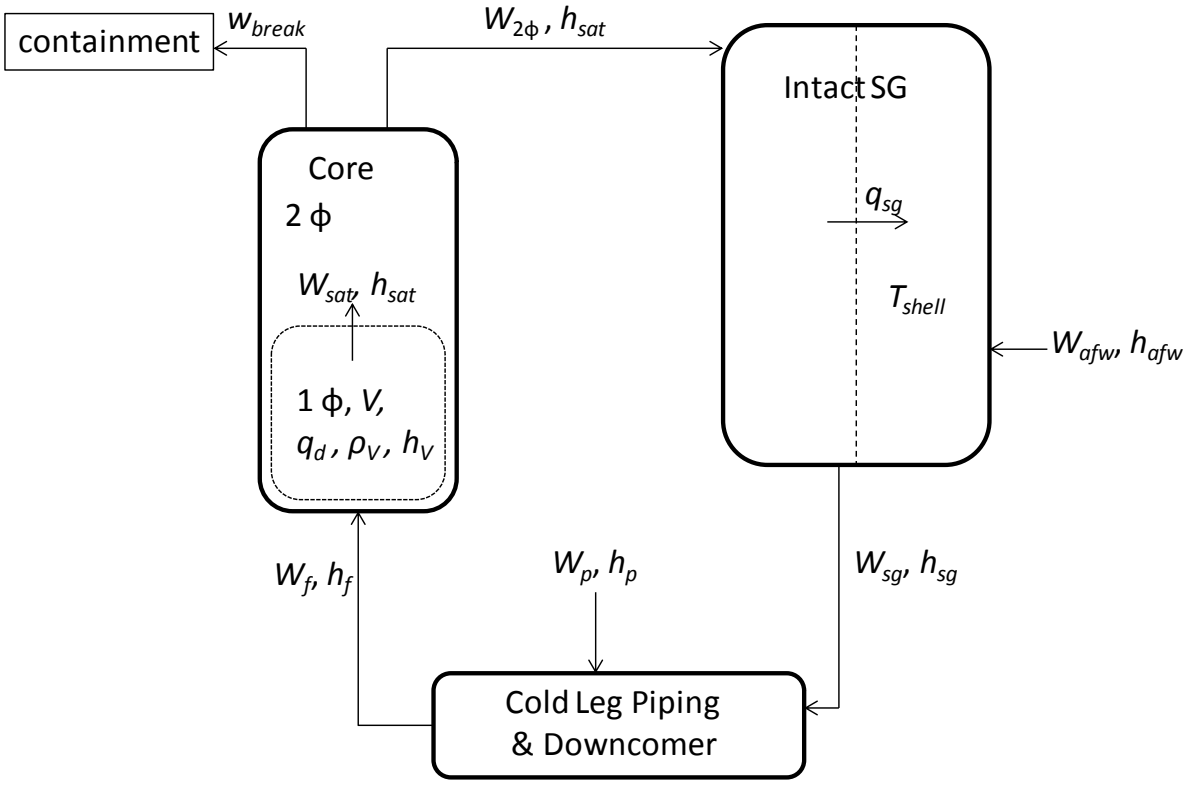

Figure 5.2. Simplified nodalization for UCN3&4 surrogate model.

The HL break is a double-ended guillotine break so the RCS remains depressurized throughout the transient. The system dynamics are further simplified because the clad and fuel temperatures are at thermal equilibrium with the fluid in the core in a quasi-steady state heat transfer regime. The most observable changes in the system are transitions between single phase flow and two phase flow in the RCS and core as a function of HPSI system function, decay power, and heat transfer in the intact SG loop. The subcooled volume V is chosen as the dependent variable of interest to parameterize the macroscopic TH behavior of the core during the recirculation phase.

To derive a functional relationship for V, we begin with the conservation of mass and energy equations for the core region of the simplified RCS nodalization

$$\rho_V \frac{dV}{dt} = W_p - W_{sat} \qquad \text{and} \tag{5.1}$$

$$h_V \rho_V \frac{dV}{dt} = q_d(V) + h_p W_p - h_{sat} W_{sat} \quad . \tag{5.2}$$

The fluid density ρ_V and specific enthalpy h_V are volume averaged properties of V. The mass flow W_p from the HPSI pumps is much greater than the mass flow W_{sg} circulating from the intact SG so the mass flow W_f into the core can be approximated as $W_f \approx W_p$ and $h_f W_f \approx h_p W_p$. Equations (5.1) and (5.2) are functions of the two unknowns V and W_{sat}. Solving Eq. (5.1) for W_{sat} and substituting into Eq. (5.2) yields

$$h_V \rho_V \frac{dV}{dt} = q_d(V) + h_p W_p - h_{sat} W_p + h_{sat} \rho_V \frac{dV}{dt}$$

$$-\Delta h_V \rho_V \frac{dV}{dt} = q_d(V) - \Delta h_p W_p \quad . \tag{5.3}$$

The subcooling enthalpy is defined as $\Delta h = h_{sat} - h$. Using the forward Euler method, Eq. (5.3) can be discretized in time to be solved for the volume at time step t_{n+1} from the parameter values at time step t_n

$$-\Delta h_V \rho_V \frac{V_{n+1}-V_n}{\Delta t} = q_d(V_n) - \Delta h_p W_p$$

$$V_{n+1} = V_n + \frac{\Delta h_p W_p \Delta t}{\Delta h_V \rho_V} - \frac{q_d(V_n)\Delta t}{\Delta h_V \rho_V} \quad . \tag{5.4}$$

Introducing the quasi-static assumption that V slowly varies with time, the time derivative in Eq. (5.3) can be set to zero yielding

$$q_d(V) = \Delta h_p W_p \quad . \tag{5.5}$$

The decay power that is generated in V is a function of the axial power distribution $f(V)$ so $q_d(V)$ in Eq. (5.5) can be separated into the time dependent term, the total decay power of the core $q_d(t)$, and $f(V)$. Equation (5.5) can yield a relationship for V as a function of the dimensionless parameter $\Delta h_p W_p/q_d$ which is similar to the Stanton number,

$$V = f^{-1}\left(\frac{\Delta h_p W_p}{q_d}\right) \quad . \tag{5.6}$$

The results of Eqs. (5.1) through (5.6) have been derived using many assumptions and simplifications so they only provide a qualitative description of the system. However, these equations suggest that over a time step, a general recursive relationship F exists between V_{n+1} and V_n and the other independent variables describing HPSI system and decay power

$$V_{n+1} = F\left(q_d, \Delta h_p W_p, W_p, \frac{\Delta h_p W_p}{q_d}, V_n\right) \quad . \tag{5.7}$$

The variables in Eq. (5.7) represent a set of input parameters to be used in surrogate training. The decay power and HPSI system variables are known or assumed whereas the heat transfer in the intact SG loop and mixing in the CLs and downcomer cannot be obtained without running RELAP5 simulations. The ACE algorithm can be used to learn the recursive relationship F from RELAP5 simulations of the LBLOCA with variations in HPSI flow rate W_p and containment sump temperature determining Δh_p. The contributions of the phenomena in the intact SG loop,

CL piping and downcomer to F are not explicitly represented in the input parameters but are accounted for in the RELAP5 simulation data from which V_{n+1} and V_n are obtained.

5.2.2 Training Set Generation and ACE Surrogate Construction

The RELAP5 UCN3&4 model was used to simulate thirteen sequences of the recirculation phase of the HL-LBLOCA to generate a training set. The HPSI flow rate was varied from 100% flow corresponding to both HPSI pumps operating at maximum flow rate conditions to degraded flow at 25%. The containment sump water temperature which determines the subcooling enthalpy of the HPSI flow was varied from 300 K, the temperature of the RWST water to 370 K which is close to saturation temperature of the RCS. Table 5.1 is the design matrix listing the HPSI flow rate and sump temperature for each sequence. Sequences 12 and 13 used a variable sump temperature given by the ramp function shown in Figure 5.3. The HPSI flow rates and sump temperatures were chosen to reflect the spectra of possible containment sump conditions that would affect the HPSI system performance during recirculation. Figure 5.4 shows the subcooled water level for the RELAP5 sequences. The subcooled water level is normalized to the height of the top of the active fuel in the core.

For each sequence, V_{n+1} and V_n were extracted at 20 s intervals from the RELAP5 output. A 20 s time step is orders of magnitudes larger from the RELAP5 time steps which are on the order of hundredths of a second. However, V_{n+1} does not vary from V_n by more than a few percent over the 20 s time steps. A decay power curve $q_d(t)$ is available from any RELAP5 sequence so $q_d(t_n)$ can be interpolated for any time step. With the assumed HPSI flow rates and sump temperatures, the thirteen sequences provide 15916 data points representing realizations Eq. (5.7). In the context of the component state and system state phase space shown in Figure 5.1, each data point is a step change in the system state dimension represented by V_n and V_{n+1} at a position in the component state space defined by $\{q_d(t_n), \Delta h_p W_p, W_p, \Delta h_p W_p/q_d\}$. In Eq. (5.7), F is the nonlinear mapping of $V_n \rightarrow V_{n+1}$ in the phase space.

Figure 5.5 show the scatterplots V_{n+1} as a function of each independent variable and time. The scatterplots reveal very little information about the functional form of Eq. (5.7) except the strong linear correlation between V_{n+1} and V_n which is expected from a recursive relationship.

Table 5.1. Design matrix of HPSI flow rate and containment sump temperature variation for RELAP5 simulations of recirculation phase HL-LBLOCA.

Case #	HPSI Flow Rate	Sump Temperature (K)	Case #	HPSI Flow Rate	Sump Temperature (K)
1	100%	300	8	50%	300
2	100%	325	9	50%	325
3	100%	350	10	50%	350
4	75%	300	11	25%	300
5	75%	325	12	100%	ramp
6	75%	350	13	50%	ramp
7	75%	370			

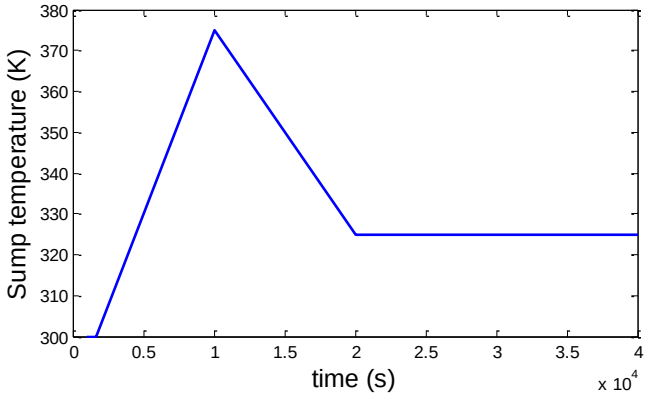

Figure 5.3. Containment sump temperature curve for ramp cases in design matrix.

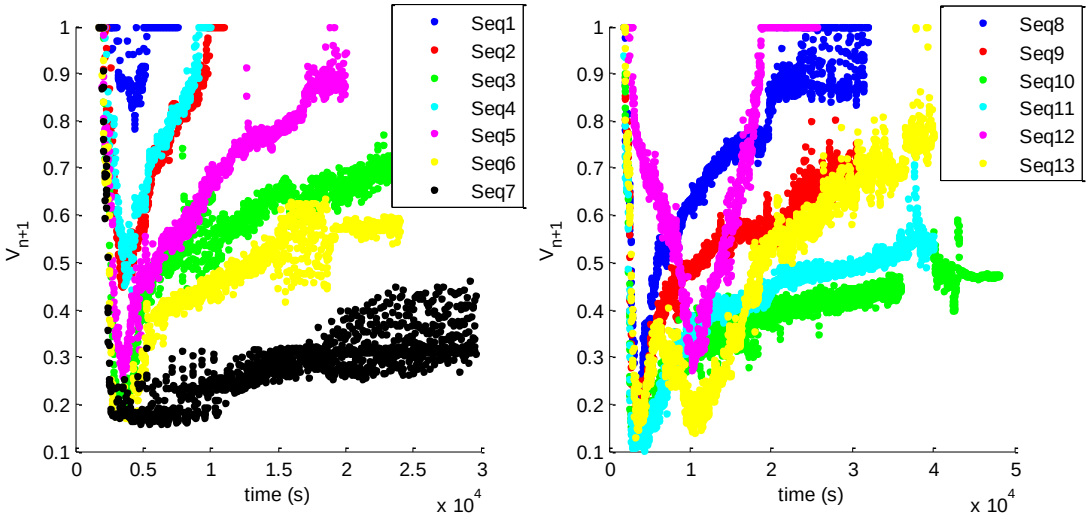

Figure 5.4. Subcooled water level for RELAP5 HL-LBLOCA sequences.

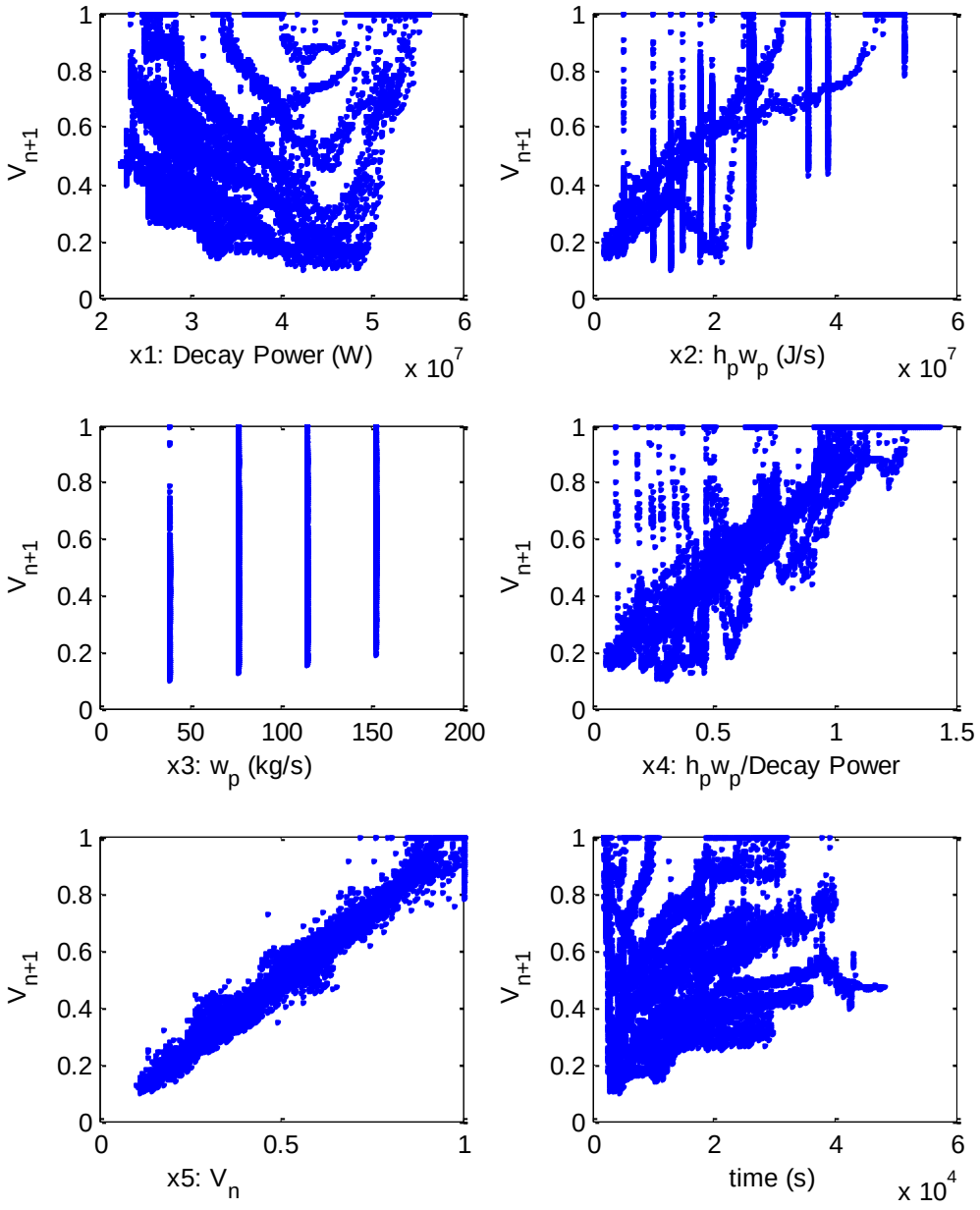

Figure 5.5. Scatterplots of V_{n+1} from RELAP5 data.

The ACE algorithm is applied to the data set to obtain the transformations shown in Figure 5.6. In the transform space, the input parameter transformations are slowly varying and smooth functions. The functional relationship of Eq. (5.7) is given by the ACE transformations

$$\theta(y) = \sum_{i=1}^{5} \phi_i (x_i)$$

$$\theta(V_{n+1}) = \phi_1(q_d) + \phi_2(\Delta h_p W_p) + \phi_3(W_p) + \phi_4\left(\frac{\Delta h_p W_p}{q_d}\right) + \phi_5(V_n) \quad . \qquad (5.8)$$

Since we are trying to generate a surrogate for the discrete time dynamic system described by Eqs. (3.59) and (5.7) and the recursive relationship between V_{n+1} and V_n, the time step $\Delta t = 20$ s is the relevant time parameter. The absolute transient time t_n of each data point is implicitly represented in the input parameters, although the decay power is an explicit function of time. The significance of the implicit treatment of time in the surrogate training is that data points occurring at very different absolute times in the RELAP5 sequences, are smoothed together as a function of their location in the input parameter and system state nonlinear phase space, recognizing the TH conditions in the core can be similar at different transient times within a given sequence and across sequences. For example, in Sequence 8 and 12 shown in Figure 5.4, the water level V_{n+1} passes through 0.7 at approximately 2,170 s and 14,000 s for Sequence 8 and 4,470 s and 16,210 s for Sequence 12. These data points are smoothed together during the ACE iterations to calculate $\theta(V_{n+1})$ and $\phi_5(V_n)$ even though the data points are from four absolute transient times that are thousands of seconds apart. Conversely, the HPSI flow rate for all of Sequence 8 was 50% versus 100% for Sequence 12 so the data points would not be smoothed together for the calculation of ϕ_2, ϕ_3 and ϕ_4.

Table 5.2 lists the ranges of the independent variable transformations and comparison to the range of the $\theta(V_{n+1})$ in the transform space as a measure of sensitivity or importance each variable. The range of the recursive component $\phi_5(V_n)$ of the ACE surrogate is equal to 73% of $\theta(V_{n+1})$ followed by subcooling enthalpy flow $\phi_2(\Delta h_p W_p)$ and decay power $\phi_1(q_d)$ at 33% and 23%, respectively. The consequence of the discrete time dynamic system model structure is the direct correlation between V_{n+1} and V_n evident through the transformations $\phi_5(V_n)$ and $\theta(V_{n+1})$ which are linear. The assumed 20 s time step limits the deviation of V_{n+1} from V_n within a few

percent so V_n must be the most important parameter that determines V_{n+1}. The other variables determine whether the water level increase or decreases from V_n.

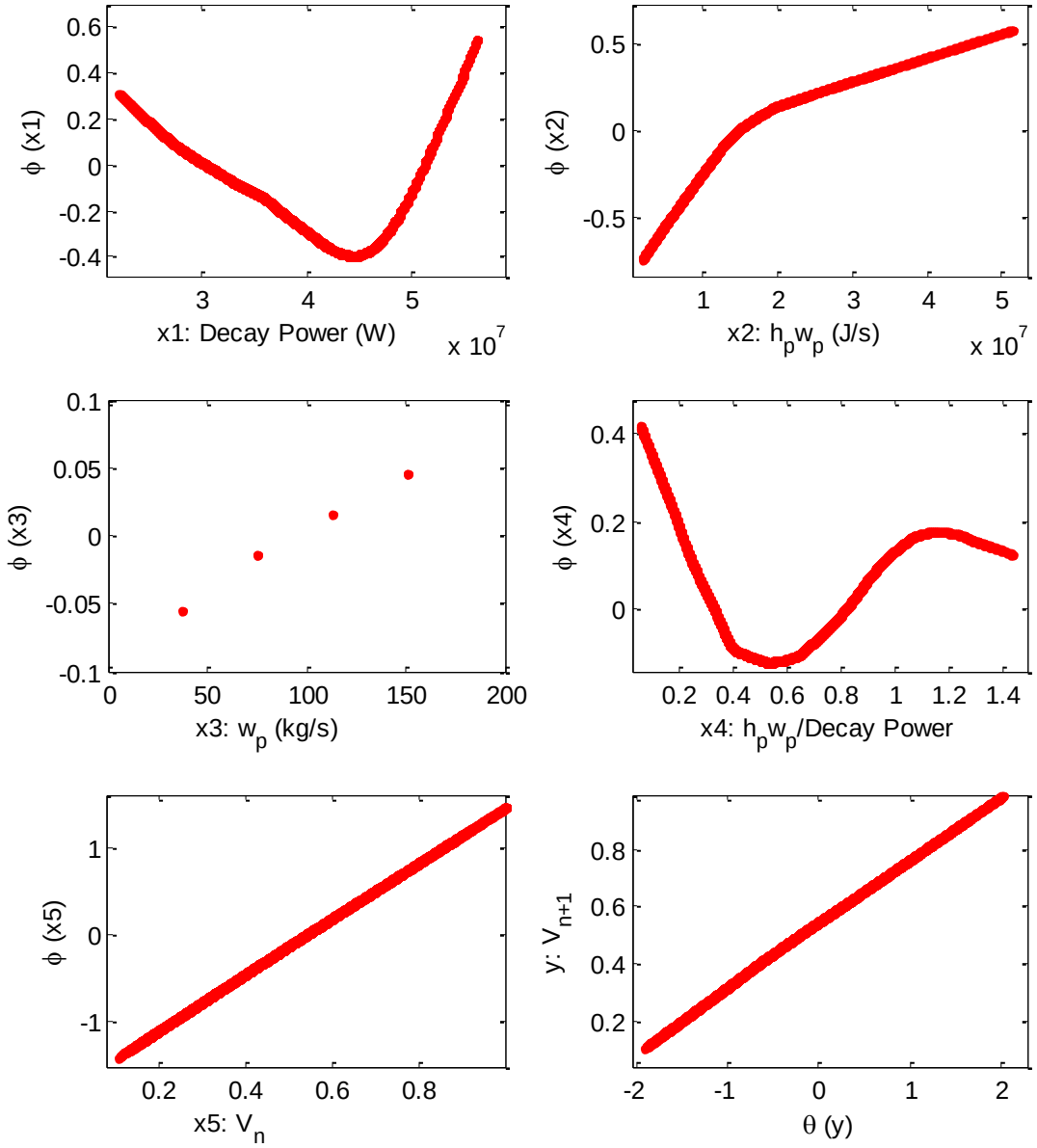

Figure 5.6. ACE transformations for subcooled water level surrogate.

Table 5.2. Range and sensitivity of ACE transformations.

	min	max	Δϕ	% of θ
θ	-1.91	2.08	3.99	1.00
ϕ_1	-0.39	0.54	0.93	0.23
ϕ_2	-0.74	0.58	1.32	0.33
ϕ_3	-0.06	0.04	0.10	0.03
ϕ_4	-0.11	0.42	0.53	0.13
ϕ_5	-1.43	1.47	2.90	0.73

5.2.3 Variance Estimate of ACE Surrogate

To quantify the model uncertainty of the ACE surrogate, we want to estimate the variance of transformed data points comprising $\theta(V_{n+1})$. Using Eqs. (3.35) through (3.37), the total variance $V[\theta(V_{n+1,j}) = \theta_j]$ for each point j is calculated from the summation of weighted variances s_{ij}^2 from each transformed input dimension i. Figure 5.7 shows the variance estimates for $\theta(V_{n+1})$ with the standard deviation calculated from $V[\theta_j]$ plotted as error bars on the transformed points. The mean variance is 0.064 corresponding to an average standard deviation of 0.25 in the transform space or approximately 5% variation in V_{n+1}. Accounting for model uncertainty, the ACE surrogate for the subcooled water level can now be expressed with a model uncertainty added as a noise term v

$$\theta(V_{n+1}) = \phi_1(q_d) + \phi_2(\Delta h_p W_p) + \phi_3(W_p) + \phi_4\left(\frac{\Delta h_p W_p}{q_d}\right) + \phi_5(V_n) + v \quad , \qquad (5.9)$$

with v distributed as $N(0, V[\theta_j])$.

Figure 5.8 shows the contributions to the variance from s_{ij}^2 for each transformed input dimension. Figure 5.8 shows that the variance from $\phi_1(q_d)$ and $\phi_2(\Delta h_p W_p)$ contribute the most to

the variance of $\theta(V_{n+1})$. This follows directly from the discussion of the implicit treatment of transient time during surrogate training. In the final ACE iteration, the smoothing operation is performed over data points with similar values of V_{n+1}. Since equivalent values of V_{n+1} from different sequences occur at different transient times, the values of $\phi_1(q_d)$ in the data subset can vary significantly as well as W_p and Δh_p between sequences. Despite having the largest range in the transform space, $\phi_5(V_n)$ contributes less to the total variance of $\theta(V_{n+1})$ because of the direct correlation between V_{n+1} and V_n. All values of $\phi_5(V_n)$ are similar in each data subset, thus having smaller variance. The range of $\phi_3(W_p)$ and $\phi_4(\Delta h_p W_p/q_d)$ are small compared to the range of $\theta(V_{n+1})$ so the contribution to the total variance from these transformed input dimensions are automatically limited.

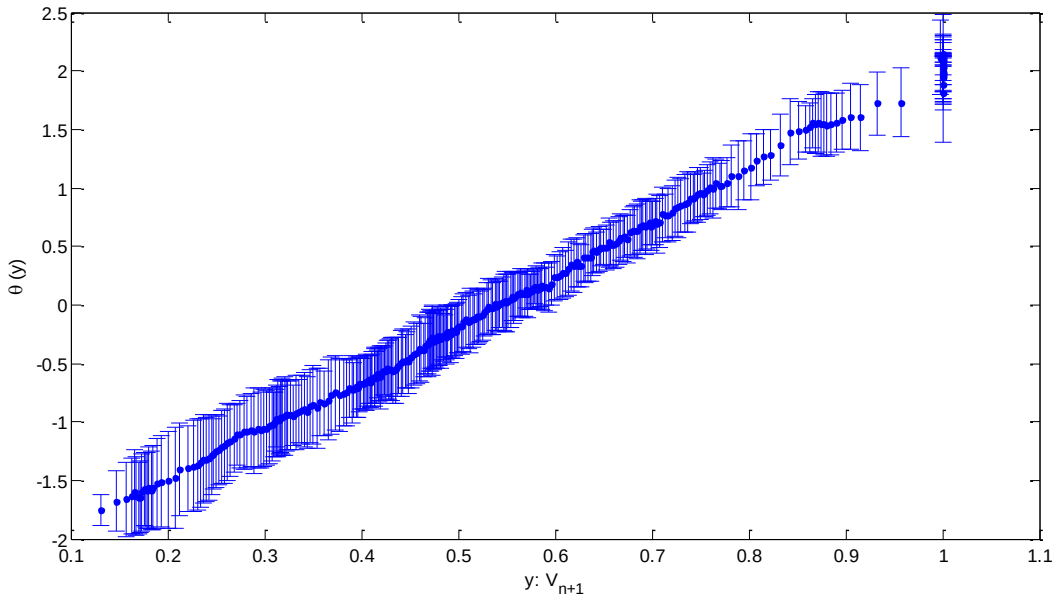

Figure 5.7. Variance estimate of $\theta(V_{n+1})$ for transformed data points.

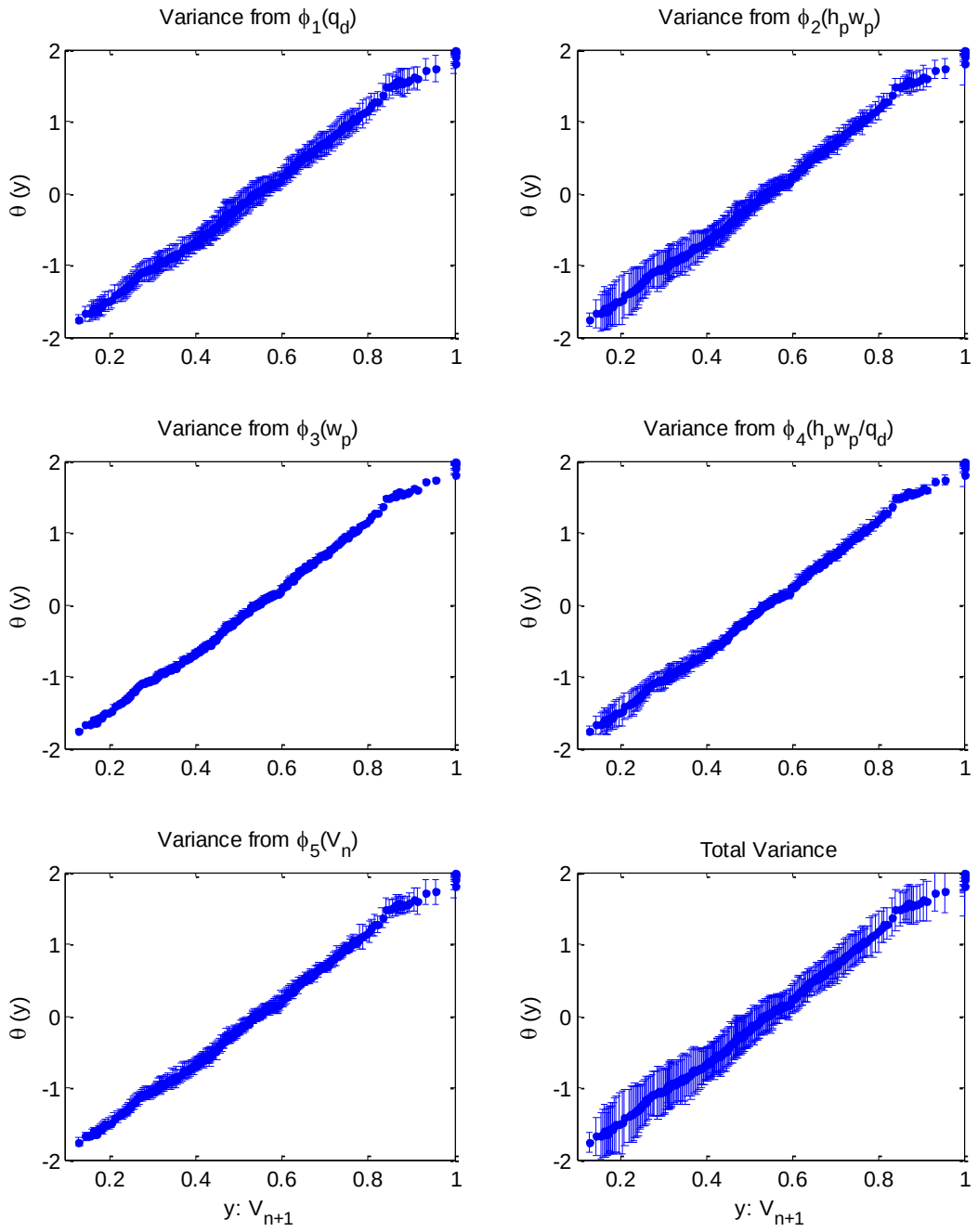

Figure 5.8. Individual contributions to variance estimate of $\theta(V_{n+1})$ from each input transformation dimension.

5.2.4 Prediction Accuracy of ACE Surrogate

To test the predictive accuracy of the ACE surrogate, the RELAP5 sequences in the training set are simulated with ACE surrogate. Two additional RELAP5 sequences, , Sequence 14 and Sequence 15, comprise a test set for cross-validation. Sequence 14 assumed 100% HPSI flow rate and a constant sump temperature of 335 K. Sequence 15 assumed 100% HPSI flow rate and a time dependent sump temperature curve shown in Figure 5.9 for the HL-LBLOCA recirculation phase obtained from the FSAR [KEP96].

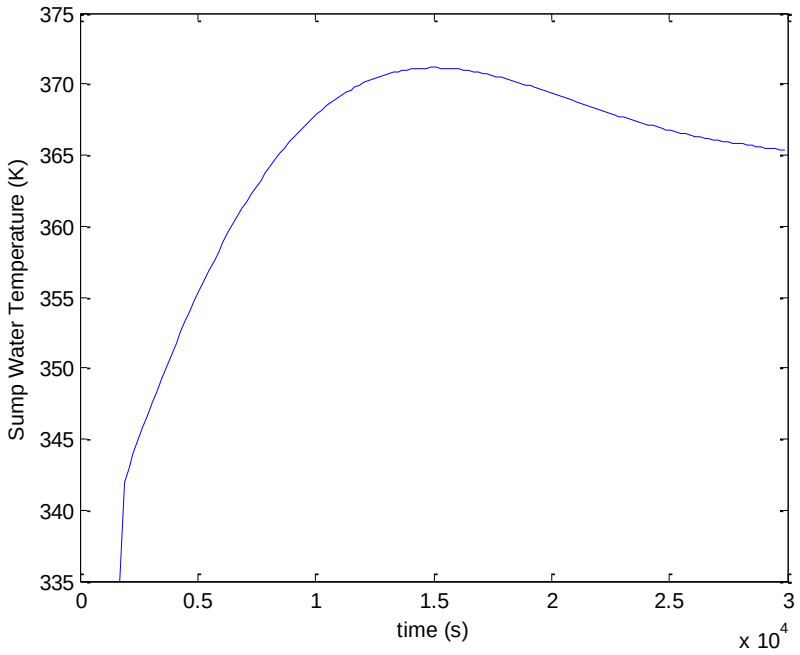

Figure 5.9. Containment sump temperature curve for HL-LBLOCA recirculation phase. From [KEP96].

Starting at a transient time of 1680 s, the time of the RAS, the ACE simulations are initialized with a subcooled water level $V_0 = 1$. Using 20 s time steps, Eq. (5.9) is evaluated and the inverse transform $\theta^{-1}()$ is taken to obtain V_{n+1}

$$V_{n+1} = \theta^{-1}\left(\phi_5(V_n) + \sum_{j=1}^{4} \phi_j(x_j) + \upsilon\right) \quad . \tag{5.10}$$

For subsequent time steps, $V_{n+1} \rightarrow V_n$ and the evaluation of Eq. (5.10) is repeated. The model uncertainty $v \sim N(0, V[\theta_j])$ introduces uncertainty into each prediction of V_{n+1} through $\theta^{-1}()$ so V_{n+1} is best described as a PDF $P(V_{n+1})$. From the recursive nature of the surrogate, the uncertainty of $V_{n+1} \sim P(V_{n+1})$ becomes the uncertainty of $V_n \sim P(V_n)$, an input variable for the next time step that must be propagated through $\phi_5()$ and $\theta^{-1}()$.

The UT is used to propagate $N(0, V[\theta_j])$ and $P(V_n)$ through Eq. (5.10) to obtain an estimate of the mean and variance of V_{n+1}. With two uncertain input parameters, the UT requires five sigma points a five evaluations of Eq. (5.10) and the UT equations given by Eqs. (3.42) through (3.44). We assume the PDF for the water level is a Gaussian defined by the mean and variance obtained from the UT, $P(V_{n+1}) = N\{\hat{V}_{n+1}, V[V_{n+1}]\}$ and $P(V_n) = N\{\hat{V}_n, V[V_n]\}$. The UT sigma points are

$$\boldsymbol{x}^0 = \begin{bmatrix} V_n^0 \\ v^0 \end{bmatrix} = \begin{bmatrix} \hat{V}_n \\ 0 \end{bmatrix}$$

$$\boldsymbol{x}^i = \begin{bmatrix} V_n^i \\ v^i \end{bmatrix} = \begin{bmatrix} \hat{V}_n \\ 0 \end{bmatrix} \pm \sqrt{(n+k)\boldsymbol{P}}\boldsymbol{Q}_i \quad i = 1, \ldots, 4 \ . \tag{5.11}$$

The covariance matrix $\boldsymbol{P}$ is

$$\boldsymbol{P} = \begin{bmatrix} V[V_n] & 0 \\ 0 & V[\theta_i] \end{bmatrix} \ . \tag{5.12}$$

The orthogonal transform $\boldsymbol{Q}$ used is

$$\boldsymbol{Q} = \begin{bmatrix} 0.7071 & 0.7071 \\ 0.7071 & -0.7071 \end{bmatrix} \ . \tag{5.13}$$

$\boldsymbol{P}$ is updated after each time step from the current estimate of $P(V_{n+1})$ and $\theta(\hat{V}_{n+1})$. For input parameters that are Gaussian distributions, $(n + k) = 3$ minimizes the error in the kurtosis so $k = 1$ is used.

Figures 5.10 and 5.11 show the fifteen RELAP5 sequences and the ACE predictions. The red curves are ACE predictions using Eq. (5.8) assuming no model uncertainty. The yellow

curves are the ACE predictions accounting for uncertainty using the UT. The mean water level $\hat{V}_{n+1}$ is plotted with error bars equal to the standard deviation from the variance estimate. The ACE surrogate appears to reproduce the RELAP5 results with reasonable accuracy for all of the sequences. The ACE prediction uncertainties estimated from the UT are on the order of 7% - 9% standard deviation for V_{n+1}. The ACE model uncertainty derived in Section 5.2.3 accounts for approximately 5% variation in V_{n+1} so it is not surprising that the total uncertainty of the prediction would be on the order of 7% - 9% when the model uncertainty is combined with uncertainty of V_n.

A noticeable difference between the ACE and the RELAP5 results is the ACE predictions are smooth curves compared to the noisy RELAP5 data. During recirculation, portions of the RCS remain voided and the localized TH conditions in the core where the single phase flow is transitioning to two phase flow display high frequency fluctuations due to the low pressure and low flow rate conditions. For example, Figure 5.12 shows the liquid fraction and pressure of control volume 110 of the UCN3&4 model, the inlet annulus of the downcomer, for Sequence 3 which affects flow in the core. The detailed RELAP5 UCN3&4 model can predict the oscillatory conditions which are evident in the water level data whereas the ACE surrogate was derived from a simplified description of the RCS. The ACE surrogate was intended to predict the macroscopic behavior of the system and was capable of learning this behavior from noisy data, a clear benefit of the ACE algorithm.

Figures 5.10 and 5.11 show that the ACE surrogate can predict the TH behavior of the recirculation phase of the HL-LBLOCA for a wide range of HPSI flow rates, containment sump temperatures and transient times. The ACE surrogate has the structure of a single equation discrete time dynamic system and takes large 20 s time steps compared to the detailed RELAP5 model of UCN3&4 that must solve thousands of coupled differential equations for hundreds of control volumes and heat structures. The ACE surrogate requires approximately 1 s of computation time to simulate 10,000 s of the recirculation phase compared to several hours of computation time for the RELAP5 model. For applications where a detailed solution for the entire NPP is not needed, a dynamic code surrogate can be an efficient alternative to a system TH code.

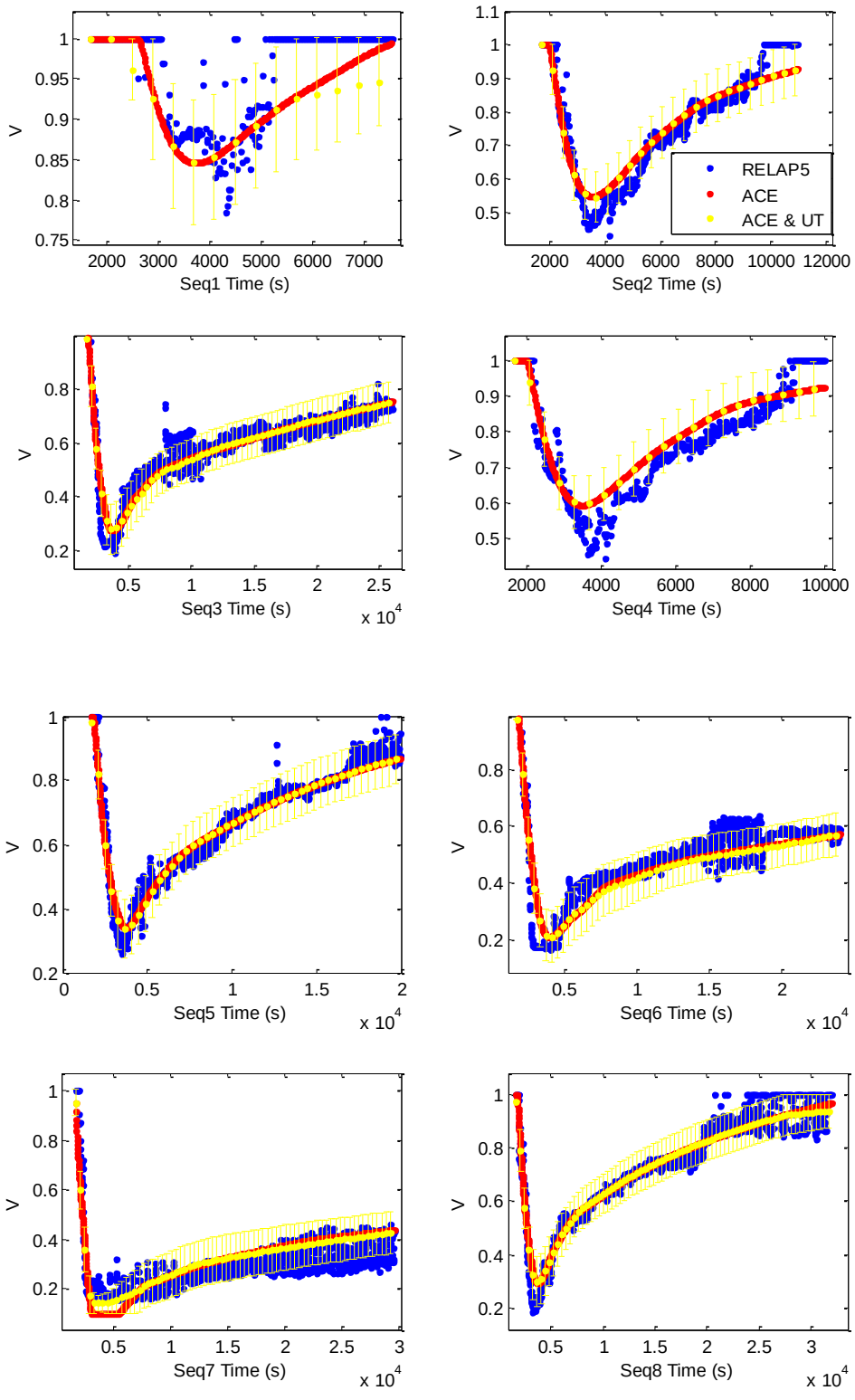

Figure 5.10. Sequences 1 - 8 for recirculation phase HL-LBLOCA.

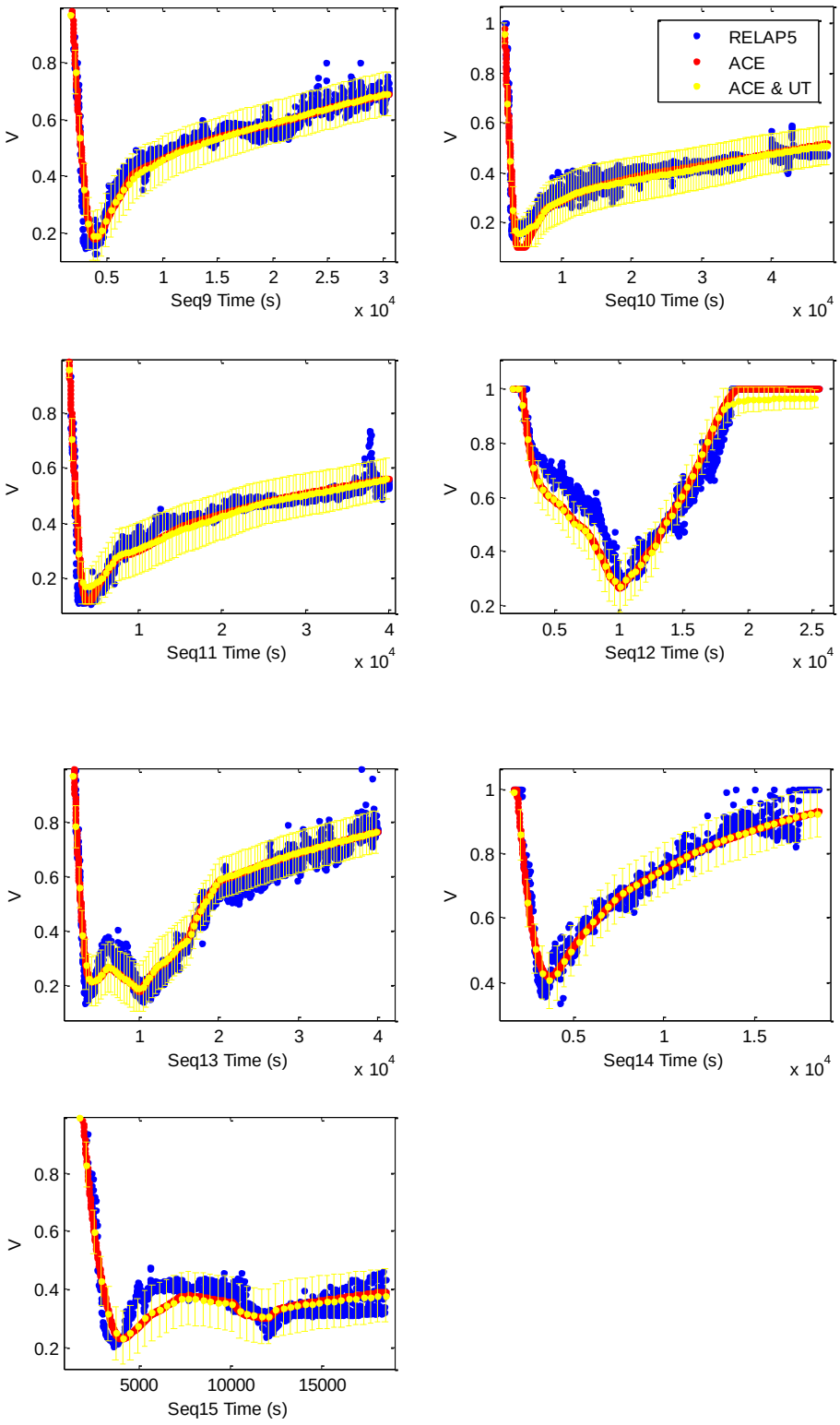

Figure 5.11. Sequences 9 - 15 for recirculation phase HL-LBLOCA.

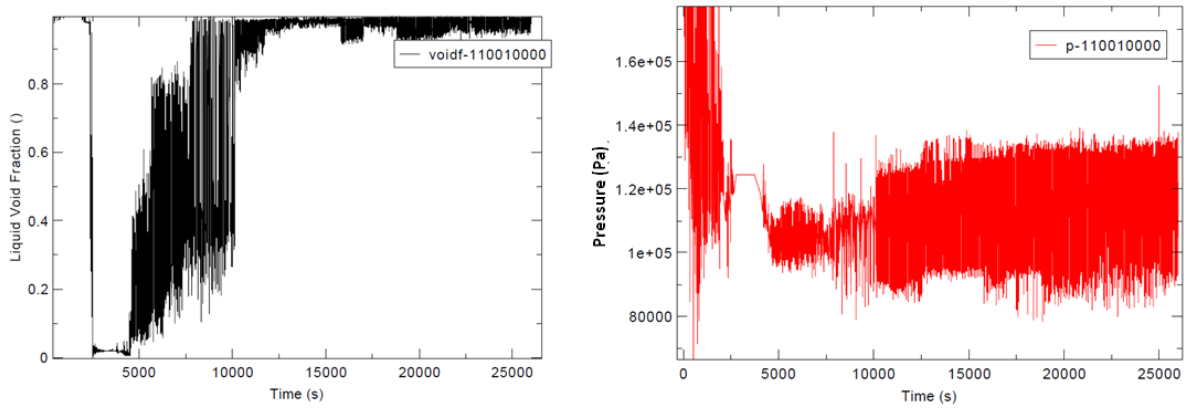

Figure 5.12. Liquid fraction and pressure in inlet annulus of downcomer for Sequence 3.

References

[KEP96] KEPCO, "Ulchin Units 3&4 Final Safety Analysis Report," Korea Electric Power Corporation, Final Safety Analysis Report, 1996.

[Mar06] M. Marsequerra, E. Zio, and R. Canetta, "Estimation of Effective Parameters in a Lumped Nuclear Reactor Model using Multiobjective Genetic Algorithms," *Nuclear Science and Engineering*, **153**, 124(2006).

CHAPTER VI

DYNAMIC EVENT TREE FOR RECIRCULATION PHASE LBLOCA

In Chapter 2, an overview of methodologies for dynamic PSA was given. While the theoretical basis for dynamic PSA methodologies such as the CET and DET is well grounded in several decades of development, the application of dynamic PSA to realistic NPP problems remain mathematically challenging and extremely computationally expensive. In this chapter, we develop two DETs for the recirculation phase of a HL-LBLOCA considering degradations of the HPSI system. The first DET models multiple degradations of the containment sump affecting the HPSI flow capacity. The second DET considers the repair time of a failed HPSI pump. The dynamic code surrogate developed in Chapter 5 is the time dependent system model used in the DET replacing direct use of the UCN3&4 RELAP5 model. The unscented transform is presented as a new method to deterministically generate both branching representing a continuous spectrum of degraded component states and branch point times requiring significantly fewer system model executions compared to conventional branching rules. The UT results are benchmarked against direct Monte Carlo generation of the DET with the surrogate model.

6.1 HPSI System Degradations During Recirculation Phase

During the recirculation phase, the two HPSI pumps draw suction from coolant water that has spilled into the two containment sumps and recirculate the coolant back to the RCS. Continued operation of the HPSI system in recirculation mode provides long term cooling to the core. The total flow delivered by the HPSI pumps is a function of the net positive suction head (NPSH) at the pump inlets and the primary system pressure in the discharge legs of the cold leg piping. Figure 6.1 shows the NPSH curve for the HPSI pumps providing the total developed

head (TDH). Operators can also manually throttle the HPSI pumps reducing flow rate to the primary system.

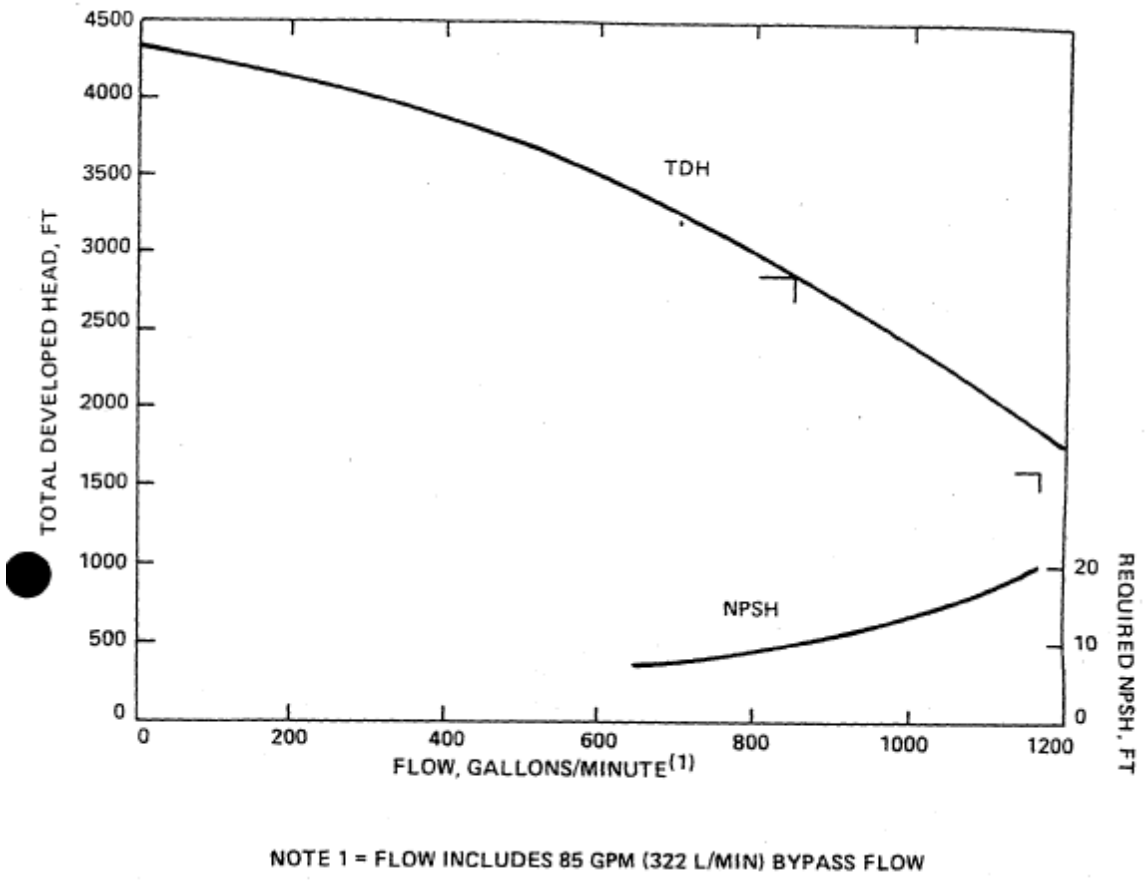

Figure 6.1. Required NPSH curve for HPSI pumps for the OPR1000. From [KEP96].

High-energy pipe line breaks such as HL-LBOCA generate debris in the containment that can accumulate on the containment sump screens. Debris accumulation is a safety issue because a clogged sump screen can reduce the available NPSH to the HPSI pumps, degrading the operability of the HPSI system and possibly failing the system completely. This safety issue has been designated Generic Safety Issue 191 (GSI-191) by the USNRC [NRC03]. Debris transport to the containment sumps during blowdown and recirculation phase of a HL-LBLOCA in the OPR1000 has been recently analyzed by CFD simulation and experimental studies [Par11,Par12]. Uncertainties in initial generation of debris types and sizes, break and sump

location, and turbulent kinetic energy models used in the CFD simulations must be considered when calculating head loss at the sump screens due to debris transport. These uncertainties suggest the consequence of the generation, transport, and accumulation of debris on HPSI system function is an uncertain and time dependent process so HPSI system degradation due to head loss at the sump screen should be considered in a probabilistic manner in dynamic PSA of recirculation phase of LOCAs.

6.1.1 Joint Probability Statement for System and Component States

The CET given by Eq. (2.1) is the joint PDF $p(x,c,t)$ for the combined system state x and component state c at time t. In the implementation of the CET [Dev92a], x and t are treated as continuous variables and the component states c are binary or ternary systems. For example, a pump is assumed to be working at *nominal* conditions or failed and a valve operates as demanded, fails closed, or fails open. Binary and ternary component systems are also assumed in DET applications [Klo06]. Discrete failure rates or transitions rates between failed states are defined for each component state and the CET or DET can be evaluated.

For the recirculation phase, the system state is the subcooled water level V and the component state is the NPSH of the HPSI pumps. Through the curve in Figure 6.1, the NPSH determines the HPSI flow rate W_p which appears in input parameters in the ACE surrogate. Here the component state is not a binary or ternary system. The HPSI system can operate in a continuous spectrum of functional conditions from the minimum flow rate to maximum flow rate depending on the NPSH. In a conventional CET or DET approach, the NPSH curve would be divided into intervals representing discrete component states noting if a higher refinement is used for a more accurate representation of the system, the curse of dimensionality rears its ugly head. We would like to avoid discretizing c and continue to treat c as a continuous variable.

Now we recast Eq. (2.1) with simplifications and assumptions specific to the recirculation phase problem. As in a DET formulation, we assume a time step $\Delta t = t - t'$ to be used in the analysis over which component transitions can occur. The containment sump conditions are independent of the TH behavior in the core so the generic component transition probability $W(c',c)$ is independent of x. $W(c',c)$ also implicitly includes the no-transition probability $W(c',c')$ so the Γ terms in Eq. (2.1) can be dropped. Equation (2.1) over a time step is

$$p(x,c,t) = \Delta t \int dx' \int dc' p(x',c',t') \delta[x - g(x',c,t-t')] W(c',c) \quad . \tag{6.1}$$

The system equation $x(t) = g()$ is the ACE surrogate model for V and $x(t)$ is uniquely determined by $c(t)$ and $W(c',c)$ resulting in

$$p[x(c),t] = \Delta t \int dc' p[x'(c'),t'] W(c' \to c) = \int dc' p[x'(c'),t'] M(c' \to c)$$

$$p_n = p[x(c_n),t_n] = \int dc_{n-1} p_{n-1} M(c_{n-1} \to c_n) = \int dc_{n-1} p_{n-1} M_n \quad . \tag{6.2}$$

Equation (6.2) gives the updated PDF $p_n = p[x(c_n), t_n]$ given a component transition over t_{n-1} to t_n governed by M_n. The product of Eq. (6.2) yields the PDF p_N for an end-state trajectory $x(c_N)$ after N time steps from an initial distribution for the system state p_0

$$p_N = \prod_{n=1}^{N} [\int dc_{n-1}] p_0 \prod_{n=1}^{N} M_n \quad . \tag{6.3}$$

A MC method can be used to solve (6.3) by sampling the PDFs $\{p_0, M_1, M_2, \ldots, M_N\}$ Z times obtaining Z values of $x(c_N)$ which are tallied to obtain an estimate of p_N. Each MC sample i is $\{x(c_0^i), c_1^i, c_2^i, \ldots, c_N^i\}$ which are an initial condition and time dependent component state input variables for a deterministic calculation of $x(c_N^i)$ performed by the ACE surrogate.

Equation (2.1) and its subsequent form represented in Eq. (6.3) are a probability model for the system state treating the joint system and component state space in probabilistic terms representing the uncertainty of the system state arising from the uncertainty in component state which is governed by the component transition rules. The deterministic system equations (the ACE surrogate) maps system state trajectories in the time domain of this joint probability space. The solution of Eq. (6.3) by a MC method resembles the UQ of a computer experiment discussed in Chapter 1. There are $N + 1$ input variables $\{x(c_0), c_1, c_2, \ldots, c_N\}$ with uncertainty defined by the PDFs $\{p_0, M_1, M_2, \ldots, M_N\}$ that are propagated through a nonlinear function, the ACE surrogate. From this interpretation, any UQ method could be used to obtain p_N or estimates of the properties of p_N. Specifically, the UT can be used to obtain an estimate of the mean and variance of p_N requiring only $2N + 3$ simulations

$$E[p_N] \cong \bar{y} = \sum_{j=0}^{2(N+1)} W^{(j)} x(c_N^j)$$

$$V[p_N] \cong \sum_{j=0}^{2(N+1)} W^{(j)} \left[x\left(c_N^j\right) - \bar{y}\right]\left[x\left(c_N^j\right) - \bar{y}\right]^T \quad . \tag{6.4}$$

Each sigma point trajectory $x\left(c_N^j\right)$ is calculated with the ACE surrogate from the corresponding sigma points $\left\{x\left(c_0^j\right), c_1^j, c_2^j, \ldots, c_N^j\right\}$ deterministically chosen by the UT algorithm of Eq. (3.42). The computational efficiency of the MC sampling and UT sampling to solve Eqs. (6.3) and (6.4) are unaffected by treating the component state *c* as a continuous variable.

6.2 DET for Multiple Degradations of the Containment Sump

To demonstrate efficiency of the proposed application of the UT to dynamic PSA and DET analysis, we will study the recirculation phase of the HL-LBLOCA in the context of GSI-191 considering containment sump performance degrading HPSI system function. The MC solution of Eq. (6.3) will be the benchmark for the UT result. The containment sumps and HPSI system will be allowed to undergo multiple degradations representing cumulative effects of accumulation of debris at the sump screens over time. As in conventional DET methodologies, we assume a fixed time step Δt = 1000 s for the analysis of the problem but we account explicitly for uncertainties in the degree of component transitions. Thus, all component degradations or transitions can only occur at multiplies of Δt so branch points only occur at discrete times.

To model uncertain degradations at all time steps, a degradation factor *F* is introduced that will be applied at each time step. The factor *F* is assumed to be distributed normally about 0.95 with a standard deviation of 0.1. At each time step, the current NPSH value is multiplied by the degradation factor *F* for each pump to determine the new degraded NPSH value. The Gaussian distribution *N* [0.95, 0.1] assumes an average 5% degradation of the NPSH over each time step and allows for the possibility of an increase in NPSH. The component state at each time step is calculated

$$c_n = c_{n-1} F_n \quad \text{and} \tag{6.5}$$

$$c_n = c_0 \prod_{i=1}^{n} F_i = 17 \prod_{i=1}^{n} F_i \ (ft) \quad . \tag{6.6}$$

The transition probability for a time step can be formally stated as

$$M_n = \frac{1}{.1\sqrt{\pi}} \exp\left(- \frac{\left(\frac{c_n}{c_{n-1}} - 0.95 \right)^2}{.02} \right) \quad . \tag{6.7}$$

The sequence begins at 1,680 s, the RAS time. Eight time steps of 1,000 s each are taken. The sequence is terminated at 9,680 s. The initial NPSH at each pump inlet is assumed to the 17 ft and the sump water temperature is assumed to be constant at 350 K.

6.2.1 Monte Carlo Solution of DET

The DET is evaluated by the MC method with a sample size of 10,000 cases. For each case, the distribution N [0.95, 0.1] is sampled sixteen times to obtain eight degradation factors for each HPSI pump. The NPSH at every time step for each pump is calculated with Eq. (6.6) and corresponding flow rates for the pumps are obtained from Figure 6.1. The ACE surrogate presented in Chapter 5 is used to calculate the subcooled water level out to 9,680 s. Figure 6.2 shows the water level trajectories for 100 MC cases.

In most DET applications, a surrogate is not available and a TH code must be used to simulate the transient, thus limiting the analysis to several hundred cases. Whether the branching is determined by preset branching rules as in DYLAM or by MC sampling, the limited sample size usually requires statistical treatment of any derived results to prevent outliers from biasing any conclusions. The branch probabilities of each case can be used as weighting factors to limit bias introduced from low-probability transitions. For example, Sequence 76 identified in Figure 6.2 is obviously an outlier having undergone a single large degradation early on in the transient. The conditional branch probability of Sequence 76 is 2×10^{-8} compared to the average conditional branch probability of the 100 cases of 3×10^{-3}. However, the sample size for the DET is large and the sampling was unbiased so no special weighting needs to be applied. A benefit of MC

methods with large sample sizes is low-probability cases, which can often have high-consequences in reactor safety, are occasionally sampled.

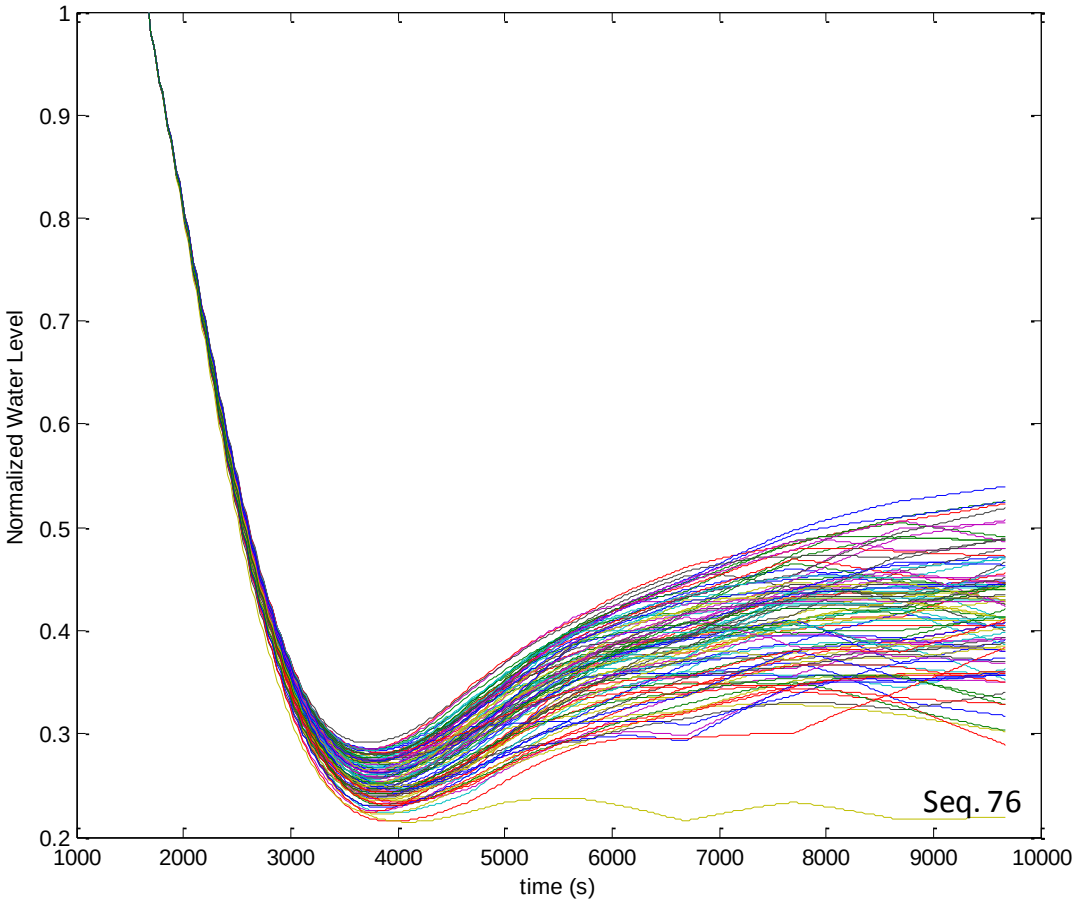

Figure 6.2. ACE surrogate simulations of 100 MC cases for the multiple degradations of the containment sump DET.

6.2.2 UT Solution of DET

The DET is evaluated by the UT method with sigma point set of 33 samples. Each sigma point is a vector of the sixteen degradation factor values, eight factors for each HPSI pump. The sigma points are generated using Eq. (3.42). All entries of $\bar{\boldsymbol{x}}$ are equal to 0.95, the mean of F, and the diagonal entries of the covariance matrix $\boldsymbol{P}$ are equal to 0.01, the variance of F. Similar

to the UT analysis in Section 3.5.2, $\boldsymbol{P}$ is transformed by a 16 × 16 orthogonal matrix obtained from a SVD of a randomly generated 16 × 16 matrix. The NPSH at every time step for each pump is calculated with Eq. (6.6) and corresponding flow rates for the pumps are obtained using Figure 6.1. The ACE surrogate is used to calculate the subcooled water level out to 9,680 s. Figure 6.3 shows the water level trajectories for the 33 UT cases. Figures 6.4 and 6.5 show the component state trajectories of the NPSH for HPSI pump A and pump B, respectively for all 33 UT cases.

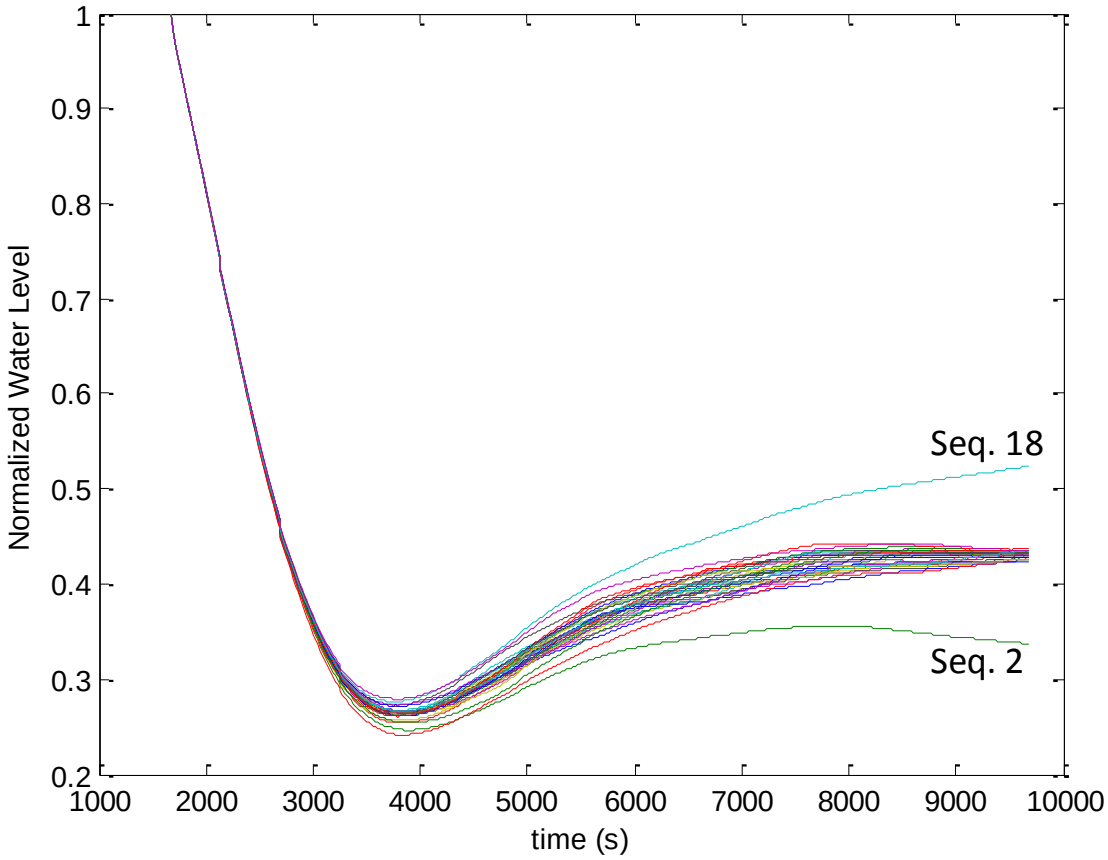

Figure 6.3. ACE surrogate simulations of 33 UT cases for the multiple degradations of the containment sump DET.

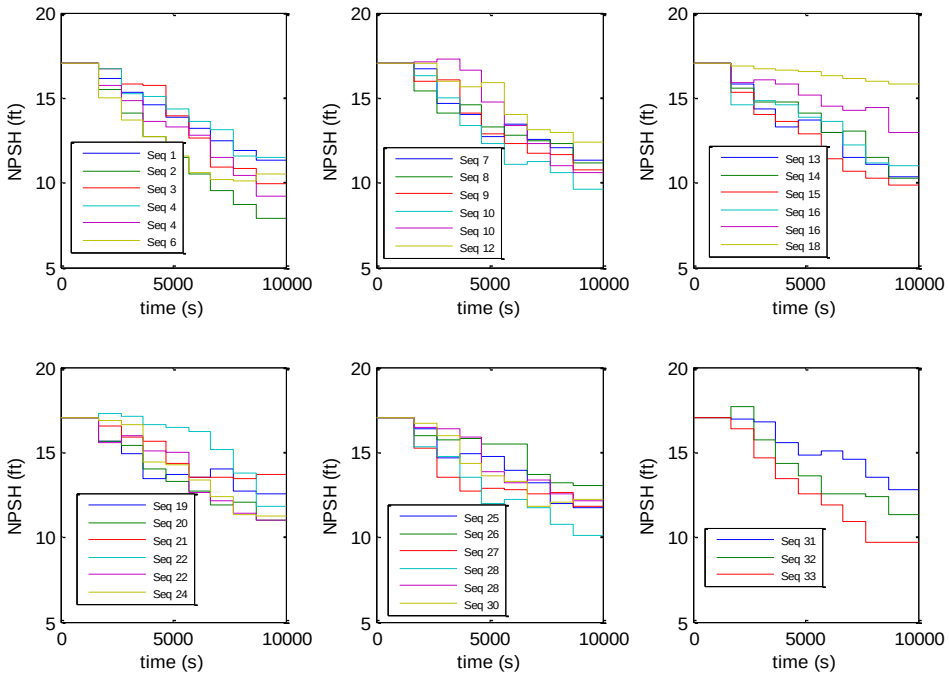

Figure 6.4. HPSI pump A NPSH degradations for UT cases.

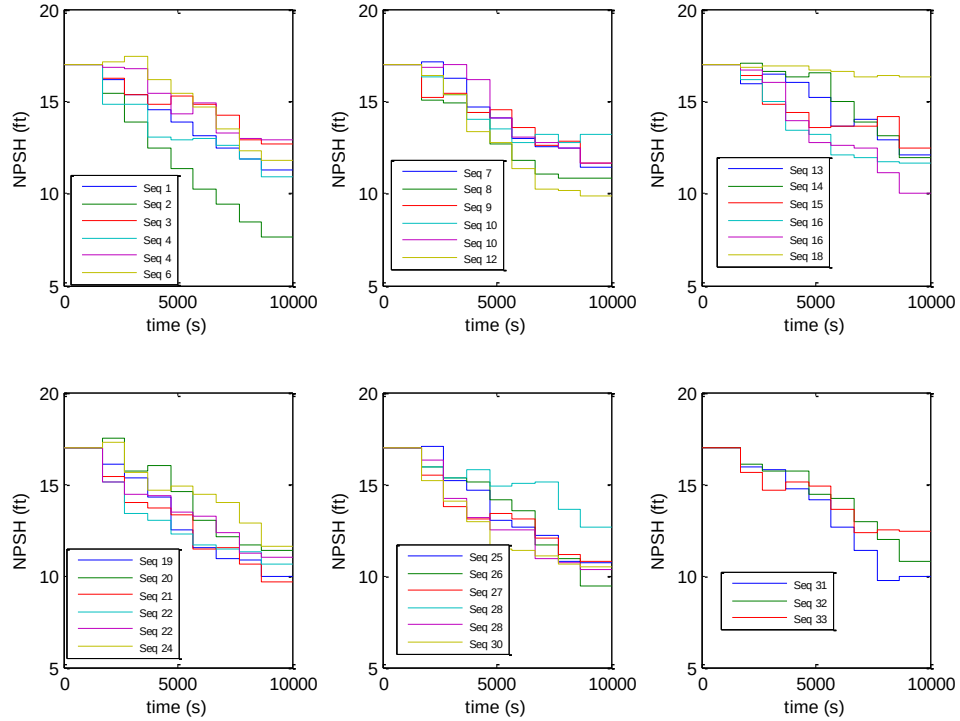

Figure 6.5. HPSI pump B NPSH degradations for UT cases.

6.2.3 Comparison of MC and UT Results

Figure 6.6 compares the mean and variance for the water level at each time step calculated from the MC and UT samples. Figure 6.7 shows the conditional PDF for water level obtained from the MC samples at each time step compared to a Gaussian distribution with mean and variance equal to UT estimates. The PDFs are conditioned on the eight pump degradation transitions. Figure 6.6 shows that the UT estimate of the mean and variance of the water level during the recirculation phase subject to multiple degradations of the HPSI systems agree very well with the MC results. Figure 6.7 shows that the true distribution of the system state at later time steps, approximated by the MC PDF, is close to a Gaussian distribution. Figure 6.7 also shows that early on in the transient over the first few time steps, HPSI system degradations affecting the HPSI flow rate do not cause very much variation in the water level. The system dynamics early on in the transient are dominated by the high decay power that decreases by almost over a factor of two over this time period. This dependence of the system state on the decay power can be seen in the ACE transformations ϕ_1 and ϕ_4 shown in Figure 5.6. The agreement of the UT, requiring only 33 samples, with MC results demonstrates the accuracy and efficiency of the UT in a DET framework.

6.3 DET for Repair of Failed HPSI Pump

For the second DET example, we consider the repair of a failed component. During a LOCA, the transition from the injection phase to the recirculation phase at the time of the RAS requires significant changes in the HPSI system configuration. The HPSI pumps suctions are automatically aligned to the containment sumps and the operators must manually isolate the empty RWST. According to the UCN3&4 EOPs, the operators must also verify that the alignment to the containment sumps was successful and the minimum NPSH requirements for the HPSI pumps are met or else the pump must be manually tripped. Unsuccessful alignment from a valve failing to open, insufficient NPSH at the sump, or inadvertent manual trip are all possible failure modes of a HPSI pump.

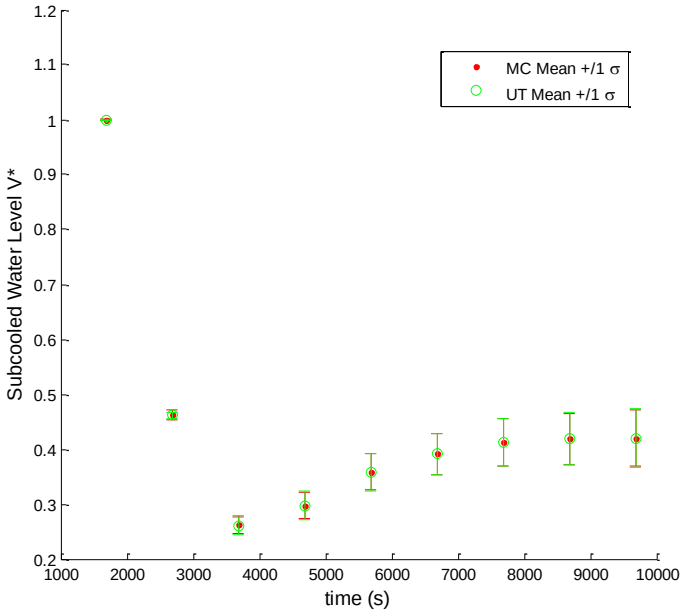

Figure 6.6. Comparison of MC and UT mean and variance estimates for the multiple degradations of the containment sump DET.

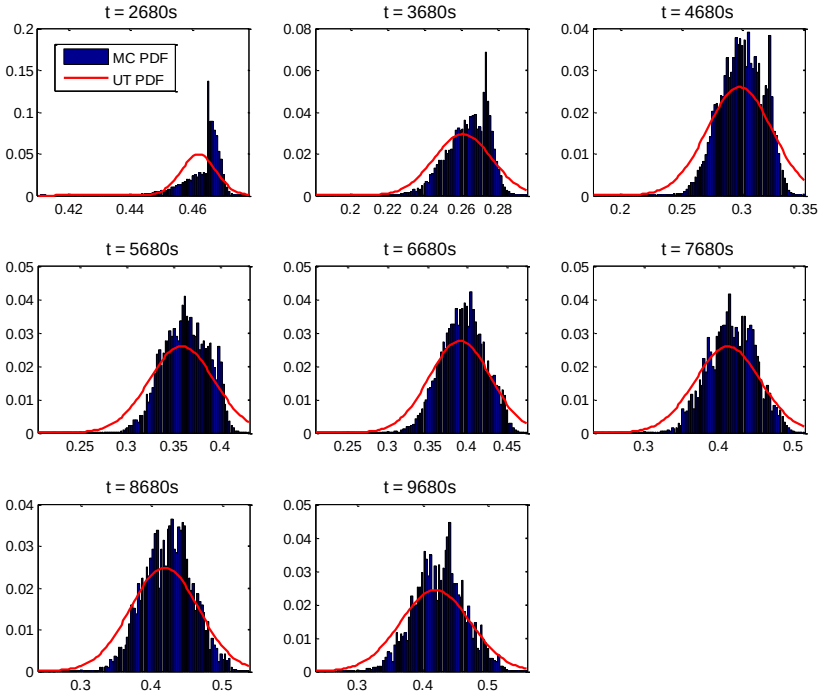

Figure 6.7. Conditional PDFs of water level for the multiple degradations of the containment sump DET. UT PDF assumes a Gaussian distribution.

In this DET, HPSI pump B is assumed to fail at the RAS time. HPSI pump A continues to operate with available NPSH assumed to be uniformly distributed *U*[7 ft, 20 ft] reflecting possible degraded sump conditions. HPSI pump B undergoes repair with the repair time normally distributed *N*[4500 s, 1800 s]. Once repaired and operating, the available NPSH for pump B is uniformly distributed *U*[7 ft, 20 ft]. The sump water temperature is assumed to be normally distributed *N*[345 K, 10 K]. These four variables reflecting uncertain repair time, degraded sump conditions and uncertain containment conditions are the input parameters to the DET. The DET is evaluated using the MC method with 10,000 simulations and the UT to sample the input parameter distributions. UT sample requires only nine simulations. The ACE surrogate is used to calculate the water level in the core for each sequence from the RAS time out to 12,000 s. All ACE water level calculations are truncated at 0.1 representing the lower plenum below the active fuel.

Figure 6.8 shows a subset of 100 MC simulations of the DET. Figure 6.9 shows the nine UT simulations for eight cases that varied the UT parameter *k*. The UT sample size is small so

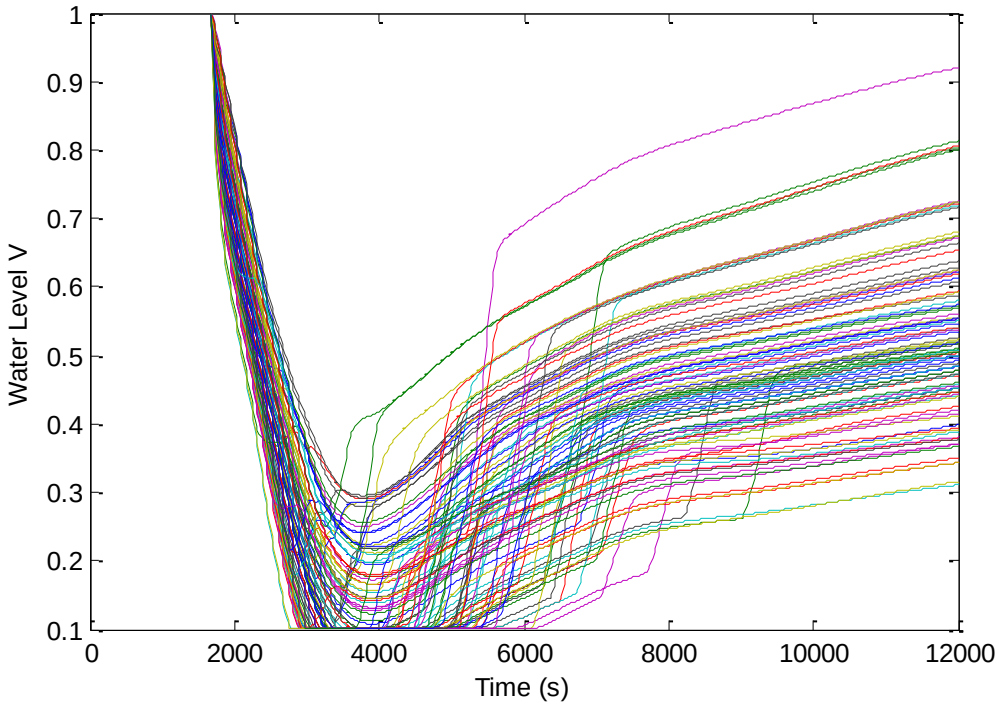

Figure 6.8. ACE surrogate simulations of 100 MC cases for the repair of HPSI pump DET.

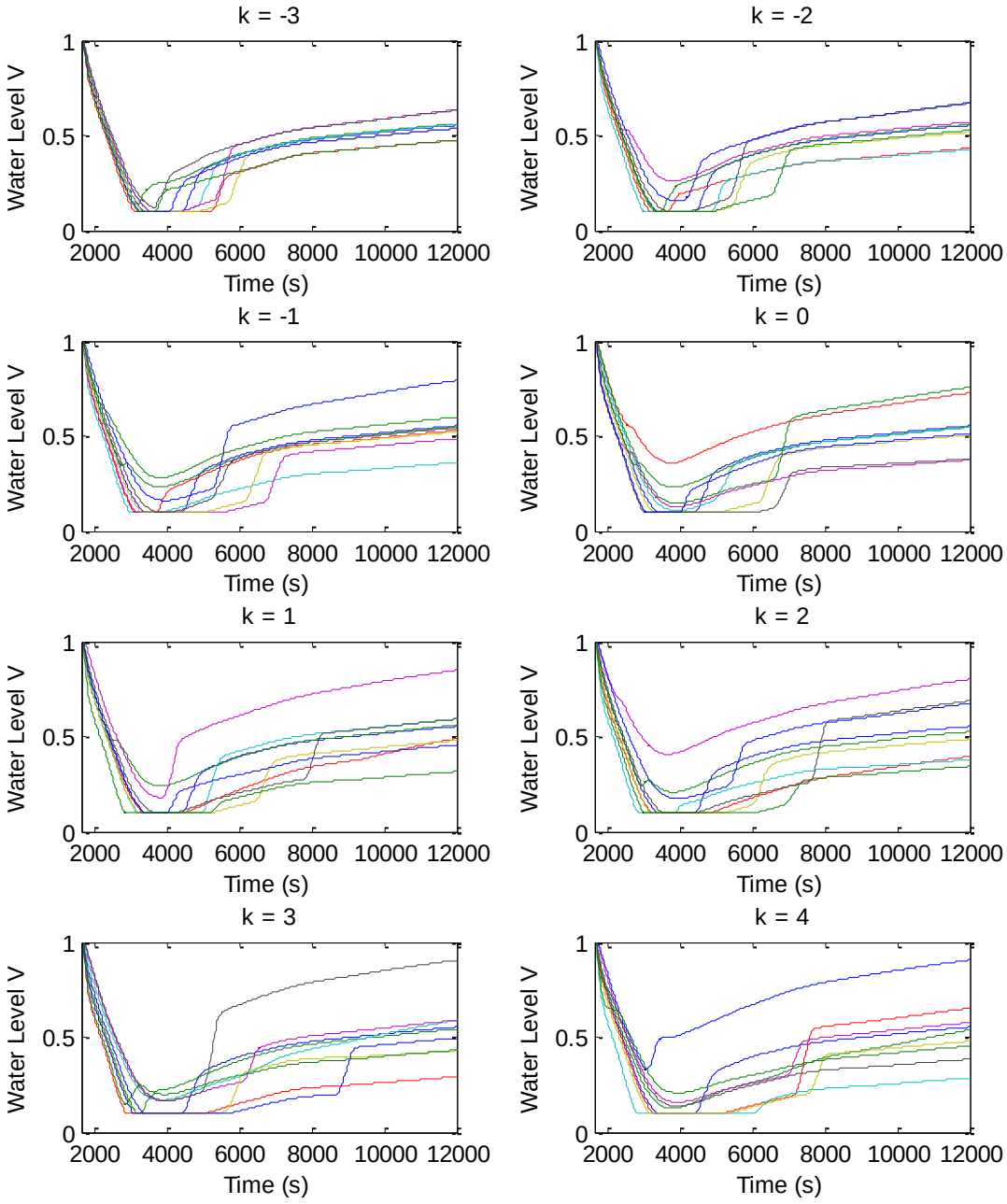

Figure 6.9. ACE surrogate simulations of UT samples varying parameter *k* for the repair of HPSI pump DET.

every simulation represents a large fraction of the total information in the sample so any derived result is sensitive to the parameter k which controls the weights and the spread of the sigma points about the central point. The central point has the weight $W^{(0)} = k/(n+k)$ and the spread of the other sigma points are determined by $\sqrt{n+k}\boldsymbol{P}$. A negative value of k moves all of the sigma points inward in the parameter space closer to the central point and assigns a negative weight to the central sigma point.

Figures 6.10 through 6.12 compares the UT mean and variance estimates to the MC results for the repair of the HPSI pump DET. The key sampled parameter is the repair time of the HPSI pump B because once the pump is operable, the total flow rate of the HPSI system can increase up to approximately 150% if pump A was operating in a degraded state. The rapid change in water level once the pump is repaired can be seen in the simulations presented in Figures 6.8 and 6.9. The mean repair time is 4500 s which is the repair time for the central point in the UT cases. For the cases with negative k shown in Figure 6.10, the effect of k on the spread of the sigma points can result in an underestimation of the variance and the large magnitude of the negative weight assigned to the central point can skew the mean prediction, especially around 4500 s when the water level of the central point simulations is rapidly changing. For $k = 0$ or k positive and small, the UT mean and variance estimates agree well with the MC results. There is some variation and disagreement between 3000 s and 6000 s when repairs are occurring and the system state rapidly changes. The asymptotic system behavior is well approximated by the UT for all cases.

6.3.1 Incorporating Model Uncertainty in DET Results

The DET results presented account for uncertainty of component states governed by component transitions but did not incorporate the uncertainty of the ACE surrogate simulations. In Chapter 5, the surrogate predictions of the RELAP5 training cases included an estimate of the prediction uncertainty which was on the order of 7% - 9% for the standard deviation of the water level. The ACE model uncertainty derived in Section 5.2.3 accounted for approximately 5% of the variation in the water level. In this section, we consider the model uncertainty of surrogate and estimate the effect on the mean and variance of the end state water level in the repair of the HPSI pump DET.

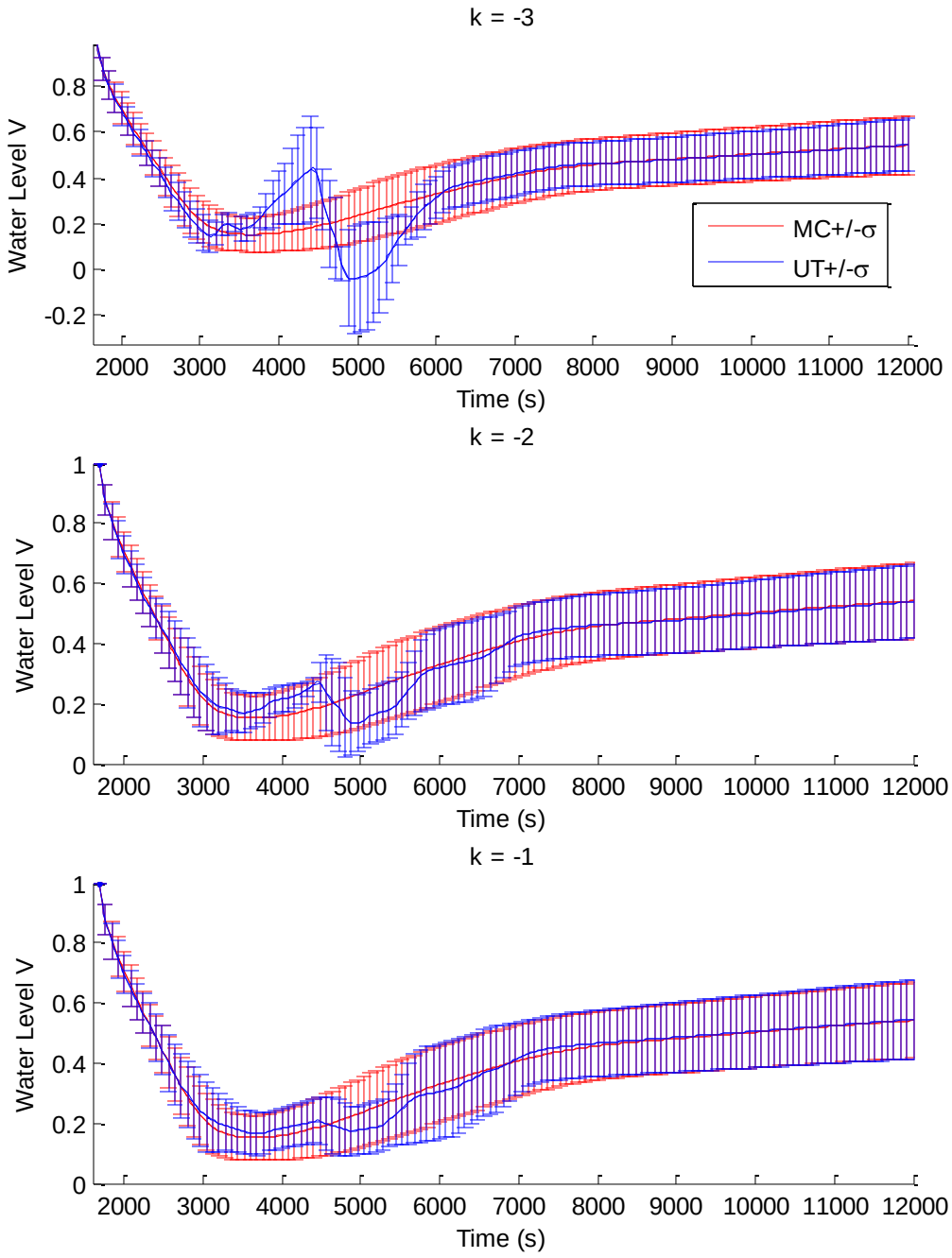

Figure 6.10. Comparison of MC and UT mean and variance estimates for the repair of HPSI pump DET.

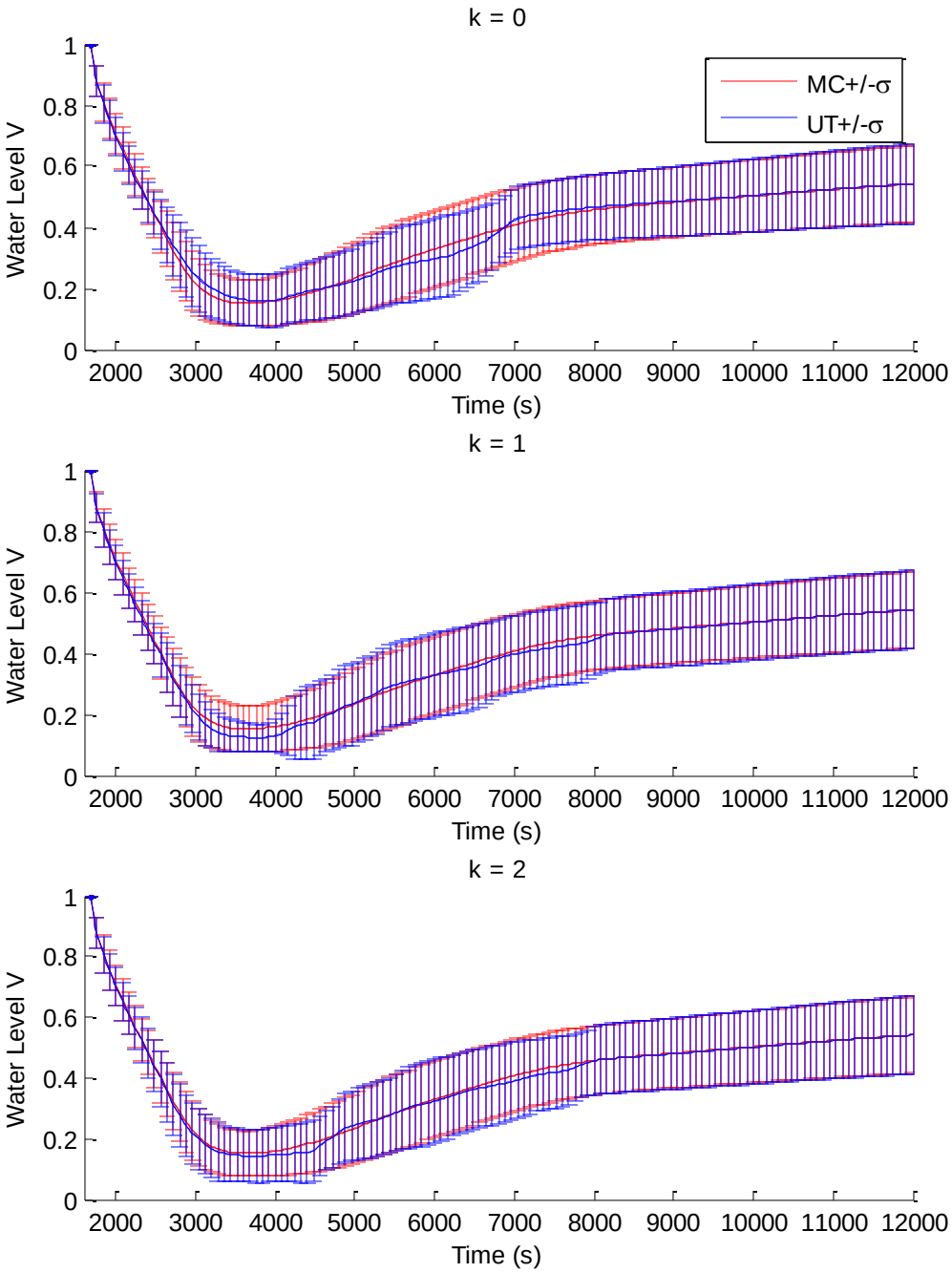

Figure 6.11. Comparison of MC and UT mean and variance estimates for the repair of HPSI pump DET.

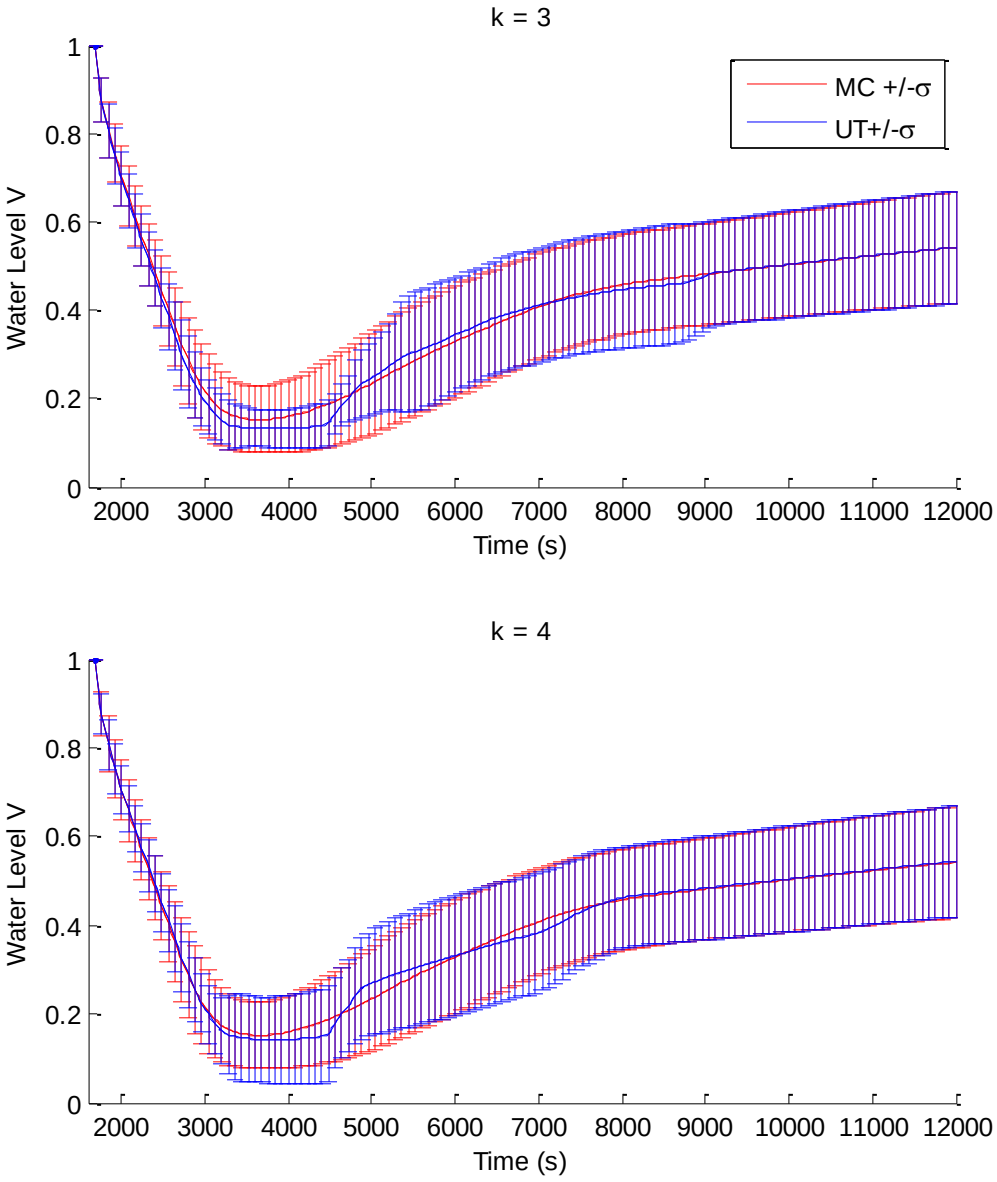

Figure 6.12. Comparison of MC and UT mean and variance estimates for the repair of HPSI pump DET.

The mean and variance estimates of the water level shown in Figures 6.10 through 6.12 where derived from the sample statistics from the MC and UT simulations assuming the data from each simulation are exact with no uncertainty for the prediction. The MC and UT sample sizes were 10,000 and 9, respectively. Considering the ACE model uncertainty, each end state

prediction is a random variable that is distributed approximately as a Gaussian with mean equal to the predicted end state and standard deviation of 0.05. The distributions should be sampled to obtain the system state values that are used in calculating the sample statistics. In the MC method, random samples can be drawn from each of the 10,000 distributions effectively increasing the size of the sample. For the UT method, the system state distribution can be treated as a fifth uncertain input parameter in the sigma point set increasing the number of simulations to 11.

Table 6.1 tabulates the mean and standard deviation for the end state water level at 12,000 s for the MC sample and UT sample for $k = 0$ and compares to the estimates accounting for model uncertainty. The MC sample with model uncertainty is 10^6 samples obtained by randomly sampling each of the 10,000 end state distributions 100 times. The mean water level is in good agreement for all samples. The UT variance estimate represented by the standard deviation agrees with the MC estimate. When model uncertainty is accounted for, the variance of the MC and UT samples increase by approximately 60% corresponding to a 26% increase in the standard deviation. For the repair of the HPSI pump DET, the component state uncertainty contributes approximately a variation $\sigma = 0.127$ in the normalized water level whereas the ACE model uncertainty contributes approximately an additional 0.035 to σ. The component state uncertainty is the dominant uncertainty contributor to the system state so the ACE surrogate, as an approximate deterministic model that adds additional uncertainty to the analysis, is acceptable to use. The results show that the ACE model uncertainty does not change the mean prediction but increases the spread of the system state samples.

Table 6.1. Mean and standard deviation estimates for water level at 12,000 s for repair of HPSI pump DET assuming 5% model uncertainty.

Water Level	MC	MC + Model Uncertainty	UT $k = 0$	UT + Model Uncertainty
mean	0.5417	0.5418	0.5425	0.5425
σ	0.1258	0.1607	0.1282	0.1622

References

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

[Dev92a] J. Devooght and C. Smidts, "Probabilistic Reactor Dynamics II: A Monte Carlo Study of a Fast Reactor Transient," *Nuclear Science and Engineering*, **111**, 241 (1992).

[Klo06] M. Kloos and J. Peschke, "MCDET: A Probabilistic Dynamics Method Combining Monte Carlo Simulation with the Discrete Dynamic Event Tree Approach," *Nuclear Science and Engineering*, **153**, 137 (2006).

[KEP96] KEPCO, "Ulchin Units 3&4 Final Safety Analysis Report," Korea Electric Power Corporation, Final Safety Analysis Report, 1996.

[NRC03] USNRC, Regulatory Guide 1.82 Revision 3, Water Sources for Long-Term Recirculation Cooling Following a Loss-of-Coolant, USNRC, Washington, DC., November 2003.

[Par11] J.P. Park, J.H. Jeong, W.T. Kim, M.W. Kim, and J.Y. Park, "Debris Transport Evalulation During the Blow-Down Phase of a LOCA Using Computational Fluid Dynamics," *Nuclear Engineering and Design*, **241**, 3244 (2011).

[Par12] J.P. Park, K.S. Choi, J.H. Jeong, G.M. Choi, J.Y. Park, and M.W. Kim, "Experimental and Numerical Evaluation of Debris Transport Augmentation by Turbulence During the Recirculation-Cooling Phase of a LOCA," *Nuclear Engineering and Design*, **250**, 520 (2012).

CHAPTER VII

SUMMARY AND CONCLUSION

Advances in computer science and widespread availability of computational resources provide unprecedented opportunities for the application of computer codes to engineering problems including safety issues affecting NPPs. In this dissertation, we developed new methodologies based on code surrogates and deterministic sampling strategies for UQ of NPP transients in reactor safety analysis. These methodologies take advantage of and efficiently use the additional computational resources available to perform more simulations with system TH codes obtaining additional and more reliable uncertainty information compared to conventional UQ methods used in reactor safety analysis. The methodologies were demonstrated for a BEPU licensing calculation and the analysis of a DET for a realistic NPP transient.

To lay the groundwork for the dissertation and motivation, an overview of system TH codes for modeling NPP transients and UQ of computer experiments was given in Chapter 1. In Chapter 2, dynamic PSA methodologies, the CET and DET, were discussed. The complexity of dynamic PSA and NPPs systems were recognized and the case for computationally efficient simulations of transients and the need to control the dimensionality of dynamic PSA problems was made. The theory of code surrogates, nonparametric regression techniques, the GPM and ACE algorithm, and the UT were presented in Chapter 3. A UQ study using GPM and ACE code surrogates representing response surfaces of the PCT during a LBLOCA and the UT as a sampling based UQ strategy was presented in Chapter 3. The LBLOCA is a classical example of UQ in BEPU methodologies used for NPP licensing and provided a benchmark for our proposed UQ methodologies. The RELAP5 TH code and the LBLOCA model for the UCN3&4 NPP was summarized in Chapter 4.

The first objective of the dissertation was to demonstrate a new methodology to develop a dynamic code surrogate that can accurately simulate time dependent, nonlinear TH behavior of a NPP transient considering multiple safety system degradations or failures. In Chapter 5, we developed an ACE surrogate that predicts the subcooled water level in the core for the recirculation phase of a HL-LBLOCA in the UCN3&4 NPP as a function of four input parameters and a recursive relationship for the water level. The dynamic surrogate is a discrete time dynamic system model and can model degradations in the HPSI system affecting the flow rate of the HPSI pumps and variation in the temperature of water in the containment sump, the SI water source during recirculation. The input parameters for the surrogate were derived from a simplified representation of the RCS and macroscopic balance equations of conservation of mass and energy. The surrogate was trained on time dependent RELAP5 data from simulations of the recirculation phase. A key feature of the surrogate is the ACE transformations represent the nonlinear phase space of the input parameters which implicitly treat the transient time of the simulations.

The second objective was to quantify the uncertainty of the dynamic code surrogate. The variance of the ACE algorithm transformations was derived in Chapter 3 providing a consistent estimation of surrogate model uncertainty. Model uncertainty was accounted for in the recirculation phase surrogate and the UT was used to estimate the uncertainty of the time dependent system state trajectory predicted by the dynamic code surrogate requiring the nonlinear transformation of model uncertainty and the system state uncertainty through the ACE model. The surrogate predictions compared to RELAP5 simulations shown in Figures 5.10 and 5.11 reveal the uncertainty band of the ACE estimates are on the order of 5 to 10% of the core height. The RELAP5 data are noisy and account for all TH processes occurring in the RCS including the intact SG loop, the CL piping, and the downcomer. The ACE surrogate only explicitly models the decay power, HPSI flow rate, and enthalpy of SI water in the input parameters so the other processes are implicitly represented in the data values for V_n and V_{n+1} used in the surrogate training. The ACE model uncertainty reflects the noise in the RELAP5 data and the TH behavior not explicitly represented in the input parameters. The subjective decision to use a code surrogate in place of the original code model is application specific and requires engineering judgment to interpret a variety of quantitative measures of the surrogate.

The model uncertainty quantified by the variance estimate is a valuable measure that can be used in this process.

The third objective was to demonstrate the accuracy of the UT as a general sampling based UQ methodology. The UT is a relatively new method and was originally developed for the specific application of extending the Kalman Filter to nonlinear system dynamics subject to model and measurement noise. In Chapter 3, the UT was applied to a BEPU calculation of a LBLOCA. The mean and variance of the PCT as function of 20 input parameters was accurately estimated by the UT. The primary advantage of the UT is the size of the UT sample determining the computational expense of the method scales *linearly* with the size of the input parameter space while guaranteeing accuracy up to the third order. Linear scaling keeps the simulation of large complex systems computationally manageable compared to geometric scaling.

The final objective of the dissertation was to address the computational challenges of direct coupling of TH codes to CETs or DETs and the curse of dimensionality in dynamic PSA of realistic NPP transients. In Chapter 6, the dynamic code surrogate replaced the direct use of the UCN3&4 RELAP5 model in a DET for the recirculation phase of a HL-LBLOCA. Multiple degradations in the high-pressure safety injection (HPSI) system and the containment sump reflecting GSI-191, a contemporary safety issue for PWRs, was studied in the DET. The UT was used to sample component transition probabilities analogous to deterministic branching rules in DET formulations. The dynamic surrogate was a computationally efficient tool allowing the MC simulation of the DET allowing the UT derived results to be benchmarked. The UT gave accurate results at significantly less cost than the MC solution.

To summarize, nonparametric regression is a powerful technique to learn functional relationships from data sets. Increased computational resources allow large training sets to be generated from best estimate computer code simulations so nonparametric regression techniques such as the ACE algorithm can be applied to reactor safety analysis. Code surrogates as response surfaces for UQ in BEPU methodologies and computationally efficient dynamic system models for dynamic PSA have been demonstrated. Some reactor safety problems will always require the high-fidelity solution of a BE code and inevitable future advances in computer science and technology may ultimately favor the direct use of the BE code in most applications. However, we have shown code surrogates can be an effective benchmarking tool for other UQ methodologies.

The UT appears to be a promising new sampling based UQ methodology with applications beyond its original implementation in the Kalman Filter. The UT is not limited to the symmetric sigma point set and associated weights that was used throughout the dissertation so the accuracy of other sigma point sets should be investigated. The mathematical subtleties of using orthogonal transforms in sigma point set generation should also be further investigated. The output PDFs in the PCT LBLOCA and DET examples presented in this dissertation were unimodal distributions with skew and kurtosis of relatively small magnitudes and the UT gave accurate estimates for the mean and variance of these distributions. The error in the UT equations are introduced in the higher order terms which are dependent on the higher order moments of the distributions. The accuracy of the UT should be tested for cases where both the input and output PDFs deviate significantly from symmetric distributions like the Gaussian and uniform. The accuracy and efficiency of the UT should also be compared to other MC based methods such as LHS that use variance reduction techniques improve the efficiency of random sampling schemes.