United States Nuclear Regulatory Commission

Protecting People and the Environment

NUREG/CR-6997
BNL-NUREG-90315-2009

Modeling a Digital Feedwater Control System Using Traditional Probabilistic Risk Assessment Methods

Final Report

Office of Nuclear Regulatory Research

United States Nuclear Regulatory Commission

Protecting People and the Environment

NUREG/CR-6997
BNL-NUREG-90315-2009

Modeling a Digital Feedwater Control System Using Traditional Probabilistic Risk Assessment Methods

Final Report

Manuscript Completed: June 2008
Date Published: September 2009

Prepared by
T.L. Chu, M. Yue, G. Martinez-Guridi,
K. Mernick, and J. Lehner, BNL
A. Kuritzky, NRC

Brookhaven National Laboratory
Upton, NY 11973-5000

A. Kuritzky, NRC Project Manager

NRC Job Code N6413

Office of Nuclear Regulatory Research

ABSTRACT

The United States Nuclear Regulatory Commission (NRC) is currently performing research on the development of probabilistic models for digital instrumentation and control systems for inclusion in nuclear plant probabilistic risk assessments. The desired goal of this research is to develop regulatory guidance for the use of risk information in regulatory decisions for new and operating reactors. This report documents the development of a reliability model of a digital feedwater control system using Markov methods supported by an automated failure modes and effects analysis (FMEA) tool. In general, the approach developed in this study should be applicable to both control and protection systems. Although the objective of the study is only to demonstrate the feasibility of the state of the art of traditional methods and data, the development of the automated FMEA tool can be considered an enhancement to the state of the art. Due to limitations in the scope of the study and the state of the art, the current model is not suitable to support regulatory decision-making. Additional research is needed to further enhance the state of the art, and potential areas of research are documented, for example, modeling of software failures.

FOREWORD

Nuclear power plants have traditionally relied on analog systems for their instrumentation and control (I&C) functions. With a shift in technology to digital systems as the result of analog obsolescence and digital functional advantages, existing plants have begun to replace some current analog I&C systems, while new plant designs fully incorporate digital systems.

The current licensing process for digital systems is based on deterministic criteria. In its 1995 Probabilistic Risk Assessment (PRA) Policy Statement, the United States Nuclear Regulatory Commission (NRC) encouraged the use of PRA technology in all regulatory matters to the extent supported by the state of the art in PRA methods and data. Though many activities are carried out in the life cycle of digital systems to ensure a high-quality product, there are no consensus methods at present for quantifying the reliability of digital systems. This has been an impediment to developing a risk-informed analysis process for digital systems.

To address this limitation, the NRC is currently performing research on the development of probabilistic models for digital I&C systems for inclusion in nuclear plant PRAs. The desired goal of this research is to develop regulatory guidance for the use of risk information in regulatory decisions for new and operating reactors. This research is consistent with the recommendations from the 1997 National Research Council report on digital I&C in nuclear power plants and with the Commission staff requirements memorandum (M061108), dated December 6, 2006, which directs the staff to address deployment of digital systems, including the area of risk-informed digital I&C.

Brookhaven National Laboratory (BNL) is supporting the NRC in this research through a project to determine the existing capabilities and limitations of using traditional (i.e., static) reliability methods to develop and quantify digital system reliability models. A previous report (NUREG/CR-6962, [Chu 2008a]) documents the initial BNL work in this area, including developing desirable characteristics for evaluating reliability models of digital systems and establishing the process for performing the reliability study of a digital feedwater control system (DFWCS) using two traditional reliability modeling methods (i.e., the event tree/fault tree method and the Markov modeling method). The current report documents the application of these methods to the DFWCS. This report also compares the resultant models to the desirable characteristics identified in NUREG/CR-6962 [Chu 2008a] to identify areas where additional research could potentially improve the quality and usefulness of digital system reliability models.

_____

Christiana H. Lui, Director
Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission

TABLE OF CONTENTS

Section		Page
ABSTRACT		iii
FOREWORD		v
LIST OF FIGURES		x
LIST OF TABLES		xi
EXECUTIVE SUMMARY		xiii
ACKNOWLEDGEMENTS		xx
ACRONYMS AND ABBREVIATIONS		xxi
1.	INTRODUCTION	1-1
1.1	Background	1-1
1.2	Objectives and Scope of Benchmark Study	1-2
1.3	Overall Approach of Benchmark Study	1-3
2.	SYSTEM DESCRIPTION AND THE SCOPE OF MODELING	2-1
2.1	Main and Backup CPUs and Their External Watchdog Timers	2-2
2.2	MFV Controller	2-5
2.3	FWP Controller	2-6
2.4	BFV Controller	2-7
2.5	PDI Controller	2-7
2.6	Other Components	2-8
3.	IDENTIFICATION OF INDIVIDUAL FAILURE MODES AND THEIR EFFECTS FOR THE DFWCS	3-1
3.1	General Issues with Current FMEAs for Digital Systems	3-2
3.2	A Generic Approach to the FMEAs of Digital Systems	3-3
3.3	FMEAs of the DFWCS Using the Generic Approach	3-5
3.3.1	FMEA of the Main CPU Module	3-6
3.3.2	FMEA of the Backup CPU Module	3-9
3.3.3	FMEA of the FWP Controller Module	3-16
3.3.4	FMEA of the MFV Controller Module	3-17
3.3.5	Considerations on the BFV Controller and the PDI Controller in the Reliability Model	3-22
3.3.6	FMEAs of Sensors, DC Power Supplies, and 120v AC Buses	3-23
3.4	Discussion and Limitations of the Generic Approach	3-25
4.	AN AUTOMATED TOOL OF PERFORMING FMEA OR DIGITAL SYSTEMS	4-1
4.1	The Advantages of Using an Automated Tool for Evaluating Failure Effects	4-1
4.2	An Automated Tool for Evaluating Failure Effects	4-2
4.2.1	Scope of the Automated FMEA Tool	4-2
4.2.2	Integrating Modules into the Automated FMEA Tool	4-3
4.2.3	Input and Output Signals of the DFWCS Modules	4-5
4.2.4	Establishing a Base Case Using Operational Data	4-12

TABLE OF CONTENTS
(Continued)

Section **Page**

4.2.5 Timing Issues Addressed in the Automated Tool 4-13
4.2.6 Criteria for Automatically Determining System Failure 4-14
4.2.7 Generation of Failure Sequences 4-15
4.2.8 Validation of the Automated FMEA Tool 4-15
4.2.9 A Summary of the Automated Tool Development and Illustrative FMEA Examples 4-16
4.3 Findings Using the Automated FMEA Tool 4-18
4.4 Discussion and Limitations of the Automated FMEA Tool 4-19
4.5 General Discussion on Developing Automated FMEA Tools for Digital Systems 4-21

5. MARKOV MODEL OF DIGITAL FEEDWATER CONTROL SYSTEM 5-1
5.1 Development of a Markov Transition Diagram 5-2
5.2 Analytical Solution of the Markov Model 5-6
5.3 A Simplified Markov Model 5-9
5.4 Discussion and Limitations of the Markov Model 5-9

6. ESTIMATION OF FAILURE PARAMETERS 6-1
6.1 Hierarchical Bayesian Analyses of PRISM Data and Failure Rates from Other Sources 6-2
6.2 Failure Mode Distributions 6-3
6.3 Common-Cause Failures (CCFs) 6-13

7. QUANTIFICATION 7-1
7.1 Quantification of Markov Model 7-1
7.2 Approximate Quantification of Markov Model 7-4
7.3 Comparison with Operating Experience 7-5

8. UNCERTAINTY ANALYSIS AND SENSITIVITY CALCULATIONS 8-1
8.1 Parameter Uncertainty 8-2
8.2 Modeling Uncertainty 8-4
8.3 Completeness Uncertainty 8-4
8.4 Sensitivity Calculations 8-6
8.4.1 Benefit of Redundancy in CPU 8-7
8.4.2 Effectiveness of Watchdog Timers 8-7
8.4.3 Benefit of Main Feedwater Valve (MFV) Demand Feedback Signals 8-8
8.4.4 Benefit of Deviation Logic 8-8
8.4.5 Summary of Sensitivity Analyses 8-9

9. COMPARISON WITH DESIRABLE CHARACTERISTICS 9-1
9.1 Level of Detail of the Probabilistic Model 9-1
9.2 Identification of Failure Modes of the Components of a Digital System 9-2
9.3 Modeling of Software Failures 9-3
9.4 Modeling of Dependencies 9-4
9.5 Probabilistic Data 9-9

TABLE OF CONTENTS
(Continued)

Section **Page**

9.6 Uncertainty 9-11
9.7 Integration of the Digital System Model with a PRA Model 9-11
9.8 Human Errors 9-12
9.9 Documentation and Results 9-12

10. COMPARISON OF RESULTS WITH THOSE FROM DYNAMIC METHODS 10-1
10.1 Application of Dynamic Methods to the DFWCS 10-1
10.2 Comparison of Scope and Level of Detail 10-2
10.3 Comparison of Results from Traditional and Dynamic Methods 10-3

11. CONCLUSIONS, INSIGHTS, AND AREAS OF POTENTIAL ADDITIONAL RESEARCH…….. 11-1
11.1 Conclusions 11-2
11.2 Insights 11-3
11.3 Areas of Potential Additional Research 11-7

12. REFERENCES 12-1

APPENDIX A FAILURE MODES AND EFFECTS ANALYSIS OF DFWCS A-1

APPENDIX B NAMING SCHEME AND COMPLETE LIST OF INDIVIDUAL FAILURE MODES B-1

APPENDIX C QUANTIFICATION OF MARKOV MODEL C-1

LIST OF FIGURES

Figure		Page
2-1	Modules of the DFWCS model	2-2
2-2	High-level flow chart of application software of each CPU	2-4
2-3	High-level flowchart of MFV controller software	2-6
3-1	Steps in the generic FMEA approach applied to the DFWCS	3-3
3-2	Major components of the main CPU module	3-6
3-3	Components of the FWP controller module	3-16
4-1	Flowchart of the automated FMEA tool	4-4
4-2	A summary of the automated FMEA tool development and implementation	4-17
5-1	Markov models for *M* independent components	5-3
5-2	Markov model of a system with *M* components	5-4
5-3	A small portion of the Markov diagram for the DFWCS	5-5
5-4	Markov model of a system with k components and each component has one failure mode	5-9
7-1	Markov diagram for and quantification of individual failure modes	7-3
7-2	Markov diagram for and quantification of both individual and double sequences	7-3
7-3	Markov diagram for and quantification of individual, double, and triple sequences	7-4

LIST OF TABLES

Table		Page
3-1	Illustrative examples of performing FMEA at component level of the main CPU module	3-10
3-2	Illustrative examples for performing FMEA at component level of the FWP controller module	3-18
4-1	Analog input signals to the main and backup CPUs	4-6
4-2	Digital input signals to the main and backup CPUs	4-7
4-3	Analog input signals to the FWP controller	4-8
4-4	Digital input signals to the FWP controller	4-9
4-5	Analog input signals to the MFV controller	4-9
4-6	Digital input signals to the MFV controller	4-9
4-7	Analog output signals of the main and backup CPUs	4-10
4-8	Digital output signals of the main and backup CPUs	4-10
4-9	Analog output signals of the FWP controller	4-11
4-10	Digital output signals of the FWP controller	4-11
4-11	Analog output signals of the MFV controller	4-12
4-12	Digital output signals of the MFV controller	4-12
6-1	Failure data used in quantifying the DFWCS reliability model	6-5
7-1	Quantification of system failure probability and frequency.	7-2
7-2	Frequency of loss of automatic control	7-4
8-1	Results of uncertainty analysis for frequency of loss of automatic feedwater control (per year)	8-3
8-2	A summary of sensitivity analyses	8-10

EXECUTIVE SUMMARY

Background

Nuclear power plants (NPPs) traditionally have relied upon analog systems for monitoring, control, and protection functions. With a shift in technology from analog systems to digital systems with their functional advantages, existing plants have begun to replace current analog systems, while new plant designs fully incorporate digital systems. Since digital instrumentation and control (I&C) systems are expected to play an increasingly important role in NPP safety, the United States (US) Nuclear Regulatory Commission (NRC) established a digital system research plan that defines a coherent set of research programs to support its regulatory needs.

Deterministic criteria underlie the current licensing basis for digital systems. In its 1995 Probabilistic Risk Assessment (PRA) policy statement, the Commission encouraged using PRA technology in all regulatory matters to the extent supported by the state of the art in PRA methods and data. At present, no methods for quantifying the reliability of digital systems are sufficiently mature to be acceptable to the NRC. Although many activities have been completed in the area of risk-informed regulation, the risk-informed analysis process for digital systems has not yet been satisfactorily developed. Therefore, one of the research programs included in the NRC's digital system research plan addresses risk assessment methods and data for digital systems.

The objective of the NRC program on risk assessment methods and data for digital systems is to identify and develop methods, analytical tools, and regulatory guidance to support: (1) using information on the risks of digital systems in NPP licensing decisions and (2) including models of digital systems into NPP PRAs. Specifically, the NRC currently is assessing the reliability of digital I&C systems, using traditional and non-traditional (dynamic) methods in parallel. For the purposes of this research, dynamic methods are defined as those explicitly attempting to model: (1) the interactions between an NPP digital I&C system and the NPP physical processes, i.e. the values of process variables and (2) the timing of these interactions, i.e., the timing of the progress of accident sequences. Traditional methods are defined here as those that are well-established but do not explicitly model either of these two aspects. An example of this type of traditional method is the Event Tree/Fault Tree (ET/FT) approach.

In the past few years, Brookhaven National Laboratory (BNL) has been working on NRC projects to investigate methods and tools for probabilistic modeling of digital systems. The work included reviewing literature on digital system modeling, reviewing and analyzing operating experience of digital systems, developing estimates of failure rates using a Hierarchical Bayesian Method (HBM) analysis, and undertaking Failure Modes and Effects Analyses (FMEAs) of digital systems. These reviews reveal that failures of digital systems have caused several events that resulted in either a reactor trip or equipment unavailability at US NPPs, at least one event at a foreign NPP that resulted in a small loss of coolant accident during refueling, and numerous significant events in other industries. Based on this experience, the potential for digital systems failures to be contributors to plant risk cannot be ruled out. The NRC tasked BNL to conduct research on using traditional reliability modeling methods for digital I&C systems, which is the subject of this report. Information on the NRC research on the use of dynamic reliability modeling methods for digital I&C systems can be found in NUREG/CR-6901 [Aldemir 2009], NUREG/CR-6942 [Aldemir 2009], and NUREG/CR-6985 [Aldemir 2009].

The principal objective of BNL's project is to determine the existing capabilities and limitations of using traditional reliability modeling methods to develop and quantify digital system reliability models. The desired goal is supporting the development of regulatory guidance for assessing risk evaluations involving digital systems. To accomplish this objective, the following tasks were performed:

1. Develop desirable characteristics for reliability models of digital systems that could provide input to the technical basis for risk evaluations for current and new reactors.

2. Select two traditional reliability methods and attempt to apply them to an example digital system to determine the capabilities and limitations of these methods.

3. Compare the resulting digital system reliability models to the desirable characteristics to identify areas where additional research could potentially improve the quality and usefulness of digital system reliability models.

In keeping with the principal objective stated above, this project generally did not involve advancements in the state of the art, such as detailed analysis and quantification of software reliability.

NUREG/CR-6962 [Chu 2008a] documents the development of the desirable characteristics for reliability models of digital systems, selection of the traditional reliability methods to be applied, and establishment of the process for performing the reliability study of a digital feedwater control system (DFWCS). As stated in NUREG/CR-6962 [Chu 2008a], the DFWCS was used since, during that phase of the project, detailed information was only available for that system. The two traditional reliability modeling methods chosen for trial application are the traditional ET/FT method and the Markov method. The former is commonly used by the US nuclear power industry and in other countries and industries. The Markov method can be a powerful tool for analyzing digital systems because it can explicitly model system configurations arising from the ability of some digital systems to detect failures and change their configuration during operation. The Markov method also explicitly treats failure and repair times. Further, the Markov method was used previously to model NPP systems and digital systems. NUREG/CR-6962 [Chu 2008a] also covers preliminary work on developing reliability models of the DFWCS, such as performing an FMEA of the system, analyzing data to estimate the failure parameters needed, and developing approaches for building Markov and ET/FT models of the system.

The current report documents the application of the selected traditional reliability methods to the DFWCS (often referred to as the benchmark study) and a comparison of the models with the desirable characteristics of NUREG/CR-6962 [Chu 2008a]. As stated above, since this project was not intended to advance the state of the art in modeling digital systems using traditional reliability modeling methods, the outcome of this project does not include identification or development of a method and supporting engineering analyses that are capable of being used for regulatory applications at the present. Rather the report identifies additional areas of research that need to be pursued in order to attain the ultimate objective of this research program. Due to these modeling limitations, as well as the weakness of publicly available digital component failure data, the current model and results cannot be used to support decision-making.

Summary of Approach

This study develops an approach for modeling digital systems and applies it to a DFWCS to demonstrate the underlying concepts of the approach. The top event selected for this proof-of-concept study is the loss of automatic feedwater control. A FMEA was performed at a relatively fine level of detail, e.g., at the level of multiplexers (MUXs) and analog/digital (A/D) converters. This level of detail is considered appropriate for supporting the proof-of-concept reliability analysis of the DFWCS. A simulation tool was developed that reflects the execution of the DFWCS software. The simulation tool is used to determine the system response to postulated hardware failure modes and combinations thereof. The important role of the simulation tool in determining system success or failure reduces the ET/FT and Markov models solely to means for quantifying system reliability (i.e., the ET/FT and Markov models are not used to identify the system failure paths, they are only used to quantify them). Since it was determined during the study that the order of component failures is important, ultimately only the Markov method was used for quantification. The sequences of component failure modes that lead to a system failure, as determined by the simulation tool, were used in defining the sequences of transitions in a Markov model. The Markov model was quantified to estimate the annual frequency with which a loss of automatic control of feedwater takes place, and to support sensitivity calculations that evaluate the benefits and importance of some of the features of the digital design. The quantification of the system model makes use of publicly available component failure parameters and the results of a HBM analysis of the raw data in the Reliability Analysis Center PRISM database that accounts for the uncertainty associated with different data sources.

The approach developed in this study, including the FMEA, simulation tool, and Markov model, should be generically applicable to digital systems. Also, while it is recognized that non-safety-related control systems and safety-related protection systems, such as a reactor protection system (RPS), have several significant design differences, it is believed that the insights and conclusions derived from this proof-of-concept study, which are mostly related to modeling methods, generally apply to both types of systems, unless otherwise noted.

Conclusions

The following conclusions were derived from performance of this study.

1. *The traditional method used in the study, i.e., Markov method, must be supported by strong engineering knowledge and supporting analyses of the systems being studied. A simulation model of the system is a critical tool in facilitating reliability model development.*

At the level of detail considered, the study requires a deterministic model that simulates the execution of the system software to capture the system design features, particularly those of the software, and to determine which sequences of postulated component failure modes would cause the system to fail. The simulation model is an enhancement to the state of the art[(1)] that allows the system behavior under failure conditions to be approximately accounted for in the reliability model, including not only the system control algorithms, but also the complex control logic based on the status of various signals of the controlled processes and that of the components of the system. Without the simulation tool, it would be very difficult, or even impossible, to directly develop a Markov or fault tree model that captures all of the details of the system design.

2. *The level of detail of the DFWCS model is adequate for capturing many of the system design features, while not being too complicated to be developed and solved.*

The Markov model of the DFWCS demonstrated the feasibility of the proposed approach. The level of detail of the model is consistent with that at which failure parameters are available (although the data has weaknesses, as discussed next). Even though the simulation tool does not encompass a thermal-hydraulic model of the plant, the system failure modes and sequences can be identified from information on its design. The state explosion problem of a detailed Markov model is resolved by truncating the higher order failure sequences when convergence is achieved. The usefulness of such a model is demonstrated further by performing a few sensitivity calculations that evaluate the importance of some of the digital design features, such as watchdog timers (WDTs), feedback of demand signals, and deviation logic.

3. *Failure parameters of digital components are scarce, and additional data are needed.*

The PRISM database is one of the few publicly available sources of digital component failure parameters. NUREG/CR-6962 [Chu 2008a] performed a HBM analysis of raw data extracted from the PRISM database to account for the variability in the sources of the data. The Bayesian analysis resulted in some failure parameters with very large error factors, demonstrating large variability in the data. It may be challenging to calculate meaningful failure rates for hardware components because of this large variability. The failure parameters used in this study are only to demonstrate the reliability method and exercise the reliability model. These data are not appropriate for quantifying models intended for use in supporting decision-making (e.g., regulatory decisions or design changes).

[(1)] While this project does not generally involve advancement in the state of the art, the development of a simulation model was deemed necessary to determine the feasibility of modeling digital I&C systems using traditional (non-dynamic) reliability methods.

Insights

A number of insights were obtained through performance of the DFWCS benchmark study. The key insights are summarized below.

- This study found that, for the DFWCS, the order in which component failure modes occur can affect the impact the failures have on the system. This is believed to be a generic feature of digital systems and should be captured in reliability models. The Markov method can easily account for the order in which component failure modes occur by considering different orders in different sequences. However, use of Markov quantification methods raises some issues, e.g., treatment of "non-minimal" sequences, with regard to integration with a PRA that is based on the ET/FT method.

- The model developed for the DFWCS is significantly more detailed than that of many other studies of digital systems. The experience of this study shows that it is difficult to capture the detailed interactions among the components and combinations of failures of the components using higher level modeling. It may be possible to use the detailed model of this study to develop an equivalent or approximate module level model by grouping the component failure modes of a module based on their impacts, e.g., on the input and output signals of the modules.

- Online repair is not considered to be possible for the DFWCS but may be possible for other digital systems, such as an RPS. If components can be repaired, the Markov model would have to be modified by adding transitions that represent repairs, making it much more difficult to solve. Using the simplified Markov method derived in this study, the governing equations with repair in the Laplace-transformed space can be solved analytically, and the inverse Laplace transform can be solved in the same way of solving the sequences without repair. Alternatively, it may be possible to develop a higher level model based on the detailed model and numerically solve the higher level model even if it includes repair.

- Performing the FMEA and running the simulation tool revealed two kinds of scenarios (one involving differences in signal delay times, and the other involving both central processing units [CPUs] operating in tracking mode) that represent potential weaknesses of the system design. The discovery of these scenarios, which were not identified in the plant's hazards analysis, suggests that the simulation tool potentially could serve to verify and validate the system software. Development of the simulation tool offers a capability to undertake test runs of the software and support deterministic evaluations of digital systems.

- This study did not specifically address Type I interactions (interactions with controlled processes external to the digital system), but considered Type II interactions (interactions among the components of the digital system) by studying the failure modes related to some events, such as communication between different components and multiplexing. Including plant dynamics could help capture subtle timing aspects of the performance of the DFWCS, e.g., issues associated with timing of failure sequences and the impacts of a within-the-range drifting signal. However, these issues are likely to be difficult to address even with a model of the plant included in the automated tool. In

addition, it is not clear, at present, whether the increased accuracy of modeling obtained through incorporation of a plant dynamics model would justify the increased complexity.

- The proposed approach of this study may also be capable of modeling safety related protection systems. For protection systems, it is believed that the use of dynamic methods may not offer any considerable improvements, because once a protection system is actuated, the feedback from the plant has no effect on the actuation.

- It is important that a reliability model realistically captures the fault-tolerance features of a digital system. In this study, the major fault-tolerance features include deviation logic in the application software, redundant CPUs, and independent WDTs of the CPUs. The first two features are well captured using the simulation tool. In the case of the WDTs, for each failure mode associated with a CPU module, plant information and an understanding about how the system works were used to determine if the effect of each failure mode on the module can be detected by its WDT and/or the application software. Fault-tolerant features may also be characterized in terms of "coverages" that typically represent the fraction of failures that can be detected. If fault coverage is accounted for in the failure data, then detailed models of the features do not have to be explicitly included in the reliability models.

Areas of Potential Additional Research

The experience of developing the probabilistic model of the DFWCS identified many areas of research to enhance the state of the art in modeling digital systems. They are summarized below.

- Improved approaches for defining and identifying failure modes of digital systems should be developed. Both software and hardware failure modes need to be considered. In this study, the hardware component failure modes may not be complete and placeholders were used for software failures. Research on software failure modes that can be incorporated in reliability models of digital systems is needed. A review of software failure experience in different industries would be beneficial.

- Software reliability methods for quantifying the likelihood of failures of both application and support software need to be developed, as well as methods for modeling software CCFs across system boundaries.

- Methods and parameter data for modeling self-diagnostics, reconfiguration, and surveillance, including using other components to detect failures, are needed. Fault-tolerance features are not limited to those modeled in this study. Different hardware redundancy techniques and software fault-tolerance designs can be applied to digital system designs. Incorporation of these different designs needs to be further pursued.

- Better data for hardware failures (both independent and common cause) and a break down of the failure rates by failure modes of digital components need to be collected. The research should include collection and analysis of generic manufacturer data and specific operating data.

- Use of Markov quantification methods raises some issues with regard to integration with a PRA that is based on the ET/FT method. Integration of Markov models, such as the one developed in this study, with an ET/FT PRA should be demonstrated.

- Methods for human reliability analysis (HRA) associated with digital systems need to be investigated. In general, digital upgrades at current NPPs and the designs of new reactors introduce new human system interfaces that are significantly different from those of existing plants. HRA research is needed to address these new interfaces in support of PRAs for both existing plants and new reactors.

- This study identified that it may be beneficial to include controlled processes in modeling drifting signals of a control system, but not necessarily for a protection system. It is also not clear whether the increased accuracy of modeling obtained through incorporation of a plant dynamics model would justify the increased complexity and effort required for intensive simulation. Determining if and when a model of controlled processes is necessary in developing a reliability model of a digital system should be further researched.

ACKNOWLEDGEMENTS

The authors are grateful to the United States Nuclear Regulatory Commission and external peer reviewers who commented on our draft report. We also express our appreciation to Avril Woodhead for her editorial review of the report, and to Jean Frejka and Nicole Kelly who put several versions of the report together and helped with the logistics of the project.

ACRONYMS AND ABBREVIATIONS

AC	Alternating Current
A/D	Analog/Digital
A/M	Automatic/Manual
ALWR	Advanced Light Water Reactor
ASIC	Application Specific Integrated Circuit
BFRV	Bypass Feedwater Regulating Valve
BFV	Bypass Feedwater Valve
BIOS	Basic Input/Output System
BNL	Brookhaven National Laboratory
CCF	Common-Cause Failure
CCMT	Cell-to-Cell Mapping Technique
CPU	Central Processing Unit
D/A	Digital/Analog
DC	Direct Current
DEMUX	Demultiplexer
DFM	Dynamic Flowgraph Methodology
DFWCS	Digital Feedwater Control System
EMI	Electromagnetic Interference
EPRI	Electric Power Research Institute
ET	Event Tree
FMEA	Failure Modes and Effects Analysis
FT	Fault Tree
FWP	Feedwater Pump
HBM	Hierarchical Bayesian Method
HRA	Human Reliability Analysis
HVAC	Heating, Venting, and Air Conditioning
IC	Integrated Circuit
I&C	Instrumentation and Control
IE	Initiating Event
I/O	Input and Output
I/P	Current-to-Pneumatic
ISA	Industry Standard Architecture

ACRONYMS AND ABBREVIATIONS
(Cont'd)

mA	Milliamperes
ms	Milliseconds
MFP	Main Feedwater Pump
MFRV	Main Feedwater Regulating Valve
MFV	Main Feedwater Valve
MFW	Main Feedwater System
MUX	Multiplexer
NCFC	Normally Closed, Fails Closed
NCFO	Normally Closed, Fails Open
NOFC	Normally Open, Fails Closed
NOFO	Normally Open, Fails Closed
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission
OOR	Out-of-Range
OORH	Out-of-Range High
OORL	Out-of-Range Low
PAL	Programmable Array Logic
PDI	Pressure Differential Indicating
PRA	Probabilistic Risk Assessment
PWR	Pressurized Water Reactor
PWR_ON	Power on
RAC	Reliability Analysis Center
RAM	Random Access Memory
ROM	Read-Only Memory
RPM	Reliability Prediction Method
RPS	Reactor Protection System
S/G	Steam Generator
SOKC	State-of-Knowledge-Correlation
SRS	Savannah River Site
URD	Utility Requirements Document
US	United States
WDT	Watchdog Timer

1. INTRODUCTION

1.1 Background

Nuclear power plants (NPPs) traditionally have relied upon analog systems for monitoring, control, and protection functions. With a shift in technology from analog systems to digital systems with their functional advantages, plants have begun such replacement, while new plant designs fully incorporate digital systems. Since digital instrumentation and control (I&C) systems are expected to play an increasingly important role in nuclear power plant safety, the United States (US) Nuclear Regulatory Commission (NRC) established a digital system research plan [NRC 2006] that defines a coherent set of research programs to support its regulatory needs.

Deterministic criteria underlie the current licensing basis for digital systems. In its 1995 Probabilistic Risk Assessment (PRA) policy statement [NRC 1995], the Commission encouraged using PRA technology in all regulatory matters to the extent supported by the state of the art in PRA methods and data. At present, no methods for quantifying the reliability of digital systems are sufficiently mature to be acceptable to the NRC. Although many activities have been completed in the area of risk-informed regulation, the risk-informed analysis process for digital systems has not yet been satisfactorily developed. Therefore, one of the research programs included in the NRC's digital system research plan addresses risk assessment methods and data for digital systems.

The objective of the NRC program on risk assessment methods and data for digital systems is to identify and develop methods, analytical tools, and regulatory guidance to support: (1) using information on the risks of digital systems in NPP licensing decisions and (2) including models of digital systems into NPP PRAs. Specifically, the NRC currently is assessing the reliability of digital I&C systems, using traditional and non-traditional (dynamic) methods in parallel. For the purposes of this research, dynamic methods are defined as those explicitly attempting to model: (1) the interactions between an NPP digital I&C system and the NPP physical processes, i.e., the values of process variables and (2) the timing of these interactions, i.e., the timing of the progress of accident sequences. Traditional methods are defined here as those that are well-established but do not explicitly model either of these two aspects. An example of this type of traditional method is the Event Tree/Fault Tree (ET/FT) approach.

In the past few years, Brookhaven National Laboratory (BNL) has been working on NRC projects to investigate methods and tools for probabilistic modeling of digital systems. The work included reviewing literature on digital system modeling [Chu 2004, Chu 2007, Chu 2008a], reviewing and analyzing operating experience of digital systems [Chu 2006], developing estimates of failure rates using a Hierarchical Bayesian Method (HBM) [Yue 2006], and undertaking Failure Modes and Effects Analyses (FMEAs) of digital systems. These reviews reveal that failures of digital systems have caused several events that resulted in either a reactor trip or equipment unavailability at US NPPs, at least one event at a foreign NPP that resulted in a small loss of coolant accident during refueling [Nuclear Energy Agency 1998], and numerous significant events in other industries. Based on this experience, the potential for digital systems failures to be contributors to plant risk cannot be ruled out. The NRC tasked BNL to conduct research on using traditional reliability modeling methods for digital I&C systems, which is the subject of this report. Information on the NRC research on the use of dynamic reliability modeling methods for digital I&C systems can be found in NUREG/CR-6901 [Aldemir 2006], NUREG/CR-6942 [Aldemir 2007], and NUREG/CR-6985 [Aldemir 2009].

The principal objective of BNL's project is to determine the existing capabilities and limitations of using traditional reliability-modeling methods to develop and quantify digital system reliability models. The desired goal is supporting the development of regulatory guidance for assessing risk evaluations involving digital systems. To accomplish this objective, the following tasks were performed:

1. Develop desirable characteristics for evaluating reliability models of digital systems that could provide input to the technical basis for risk evaluations for current and new reactors.

2. Select two traditional reliability methods and attempt to apply them to an example digital system to determine the capabilities and limitations of these methods.

3. Compare the resulting digital system reliability models to the desirable characteristics to identify areas where additional research could potentially improve the quality and usefulness of digital system reliability models.

In keeping with the principal objective stated above, this project generally did not involve advancements in the state of the art, such as detailed analysis and quantification of software reliability. Earlier BNL work on software reliability is summarized in [Chu 2007].

NUREG/CR-6962 [Chu 2008a] documents the development of the desirable characteristics for evaluating reliability models of digital systems, selection of the traditional reliability methods to be applied, and establishment of the process for performing the reliability study of a digital feedwater control system (DFWCS). As stated in NUREG/CR-6962 [Chu 2008a], the DFWCS was used since, during that phase of the project, detailed information was only available for that system. The two traditional reliability-modeling methods chosen for trial application are the traditional ET/FT method and the Markov method. The former is commonly used by the US nuclear power industry and in other countries and industries. The Markov method can be a powerful tool for analyzing digital systems because it can explicitly model system configurations arising from the ability of some digital systems to detect failures and change their configuration during operation. The Markov method also explicitly treats failure and repair times. Further, the Markov method was used previously to model NPP systems and digital systems. NUREG/CR-6962 [Chu 2008a] also covers preliminary work on developing reliability models of the DFWCS, such as performing an FMEA of the system, analyzing data to estimate the failure parameters needed, and developing approaches for building Markov and ET/FT models of the system.

The current report documents the application of the selected traditional reliability methods to the DFWCS (often referred to as the benchmark study). This report also includes a comparison of the models with the desirable characteristics of NUREG/CR-6962 [Chu 2008a].

1.2 Objectives and Scope of Benchmark Study

The objectives of the benchmark study documented in this report are twofold: (1) to apply two traditional methods, i.e., Markov and ET/FT methods, to a DFWCS, building on the work done in NUREG/CR-6962 [Chu 2008a] and (2) to compare the models against the NUREG/CR-6962 [Chu 2008a] desirable characteristics to evaluate the state of the art and identify areas where additional research would enhance this knowledge. As stated above, the DFWCS was selected as the initial benchmark system for the proof-of-concept study due to the availability of the necessary detailed system information. While it is recognized that non-safety-related control systems and safety-related protection systems have several significant design differences, it is believed that the

insights and conclusions derived from this proof-of-concept study, which are mostly related to modeling methods, generally apply to both types of systems, unless otherwise noted.

This proof-of-concept study models the DFWCS while the plant is operating at full power, and estimates the frequency that a loss of automatic control of the system takes place, including switchover to manual control and incorrect control output signals, caused by hardware failures of its components and support systems. Modeling manual control is beyond the scope of the study. In general, other top events associated with the DFWCS can be defined, and models of those top events can be developed accordingly. External causes of failure, such as fires and seismic events, and other modes of system operation are beyond the scope of this study. Due to the lack of consensus on software reliability methods, modeling software failures also is beyond the scope of this project, though placeholders for software failure rates are identified in the models. An arbitrarily selected small failure rate is used in the model quantification, such that the contribution of software failure does not mask the contribution from other modeled failures. The inclusion of placeholders for software failures is not intended to imply that it is appropriate to model hardware and software failures as separate entities, nor that software failures can be addressed probabilistically. These placeholders merely serve to indicate that ultimately software reliability should be addressed in some manner, even though it is out of the scope of the current study. Integration of the reliability model developed in this study with a nuclear plant PRA is also beyond the scope of this study.

As stated previously, the objective of the NRC program on risk assessment methods and data for digital systems is to identify and develop methods, analytical tools, and regulatory guidance to support: (1) using information on the risks of digital systems in NPP licensing decisions and (2) including models of digital systems into NPP PRAs. Since the principal objective of the current project was only to evaluate the existing state of the art in modeling digital systems using traditional reliability modeling methods, and not to advance the state of the art, the outcome of this project does not include identification or development of a method and supporting engineering analyses that are capable of being used for regulatory applications at the present. Rather the report identifies (in Section 11.3) additional areas of research that need to be pursued in order to attain the ultimate objective of this research program. Due to these modeling limitations, as well as the weakness of publicly available digital component failure data, the current model and results cannot be used to support decision-making.

1.3 Overall Approach of Benchmark Study

This study found that at the level of detail that is modeled, it is not possible to deductively develop ET/FT logic or identify the Markov states that represent system failure, as is usually done in traditional ET/FT and Markov analyses. Instead, an automated FMEA tool was developed to identify the sequences of failures that lead to a system failure, and Markov and fault tree methods are only considered as a means to quantify the sequences. Although an automated tool was used, the methods applied are still referred to as “traditional,” since they do not attempt to explicitly model the interactions between the DFWCS and the plant physical processes.

In this study, the approaches described in NUREG/CR-6962 [Chu 2008a] for developing ET/FT and Markov models of the DFWCS were attempted and modified, as necessary, to develop reliability models of the system. The modifications included the following:

1. The FMEA approach of the main central processing unit (CPU) module[1] developed in NUREG/CR-6962 [Chu 2008a] was applied to other modules in the system. To correctly determine the effects of the postulated failure modes, following the suggestion of NUREG/CR-6962 [Chu 2008a], a simulation model of the system was developed that includes the actual software of the modules and needed interfaces to automate the process of determining the system's responses. The simulation tool also was used to assess the system's response to combinations of failures.

2. Because the simulation tool can handle combinations of failure modes as well as individual ones, there was no need to group these modes as proposed in NUREG/CR-6962 [Chu 2008a]. This change is expected to enhance the model's accuracy.

3. Since the simulation tool generates combinations of failures that lead to system failure, the fault tree and Markov model approaches proposed in NUREG/CR-6962 [Chu 2008a] were not needed to identify combinations of failures. The generated failure combinations were used directly in quantification. In fact, at the level of detail considered in this study, it is not feasible to deductively develop a fault tree or Markov transition state model for the DFWCS.

4. The simulation tool was used to investigate the effects of the order in which failure modes occur. This investigation revealed that, in some cases, the order of failure does make a difference. Therefore, it is necessary to explicitly account for the order in which failures occur, and more appropriate to refer to the combinations of failure modes as failure (or failure mode) sequences. Since the order of the failures was found to be important, the failure sequences were quantified using a Markov model. Both an exact solution and an approximate solution to the Markov model were derived and used in quantifying the top event.

5. The sequences of the Markov model are similar to cutsets typically considered in a PRA, except for the way in which the sequences are quantified. Use of the Markov quantification method makes it more difficult to integrate the model with a PRA that is based on the ET/FT method. The integration is beyond the scope of this study. It can likely be done by converting the sequences into equivalent cutsets and using approximate methods of quantification.

The approach demonstrated in this proof-of-concept study should be applicable to any digital system. It is based on the use of failure modes of generic components of digital systems and publicly available component failure data. The level of detail of the model allows important digital design features to be captured. In particular, the use of an automated tool developed using the actual system software allows the software to be more realistically accounted for in the modeling. The use of a Markov model for quantifying the system-failure sequences takes into consideration the order in which failures occur in the sequences and the competition among failure modes of components. However, due to limitations in the state of the art for modeling digital systems, several significant issues remain to be resolved, as identified in this report.

[1] In general, CPU represents a central processing unit, which is a generic component of digital systems. Here, a CPU module includes a CPU and its associated components such as a multiplexer, analog/digital converter, etc. In this study, a CPU of a CPU module is denoted as a microprocessor in order to avoid confusion with a CPU module, and CPU and CPU module are used interchangeably to represent a CPU module.

The following summarizes the approach used in this study with references to the chapters that provide more detailed documentation.

Definition of Top Event

This study is based on a DFWCS of a two-loop pressurized water reactor (PWR). Each of the two reactor-coolant loops has a DFWCS. The top event selected for the proof-of-concept study is failure of a DFWCS to automatically control feedwater to its associated steam generator while the plant is operating at full power during one year. This can be considered a contributor to the loss of main feedwater initiating event for the PRA of the plant. The defined top event does not take into consideration the possibility of manually controlling the system. For some system failure modes, manual control is still possible using the DFWCS. However, modeling manual control is beyond the scope of the study.

The system also performs its functions during low-power mode and after a reactor trip; these functions are beyond the scope of the study. Chapter 2 gives a summary description of the system and defines the system boundary of the modeling performed in this study; a more detailed description is provided in NUREG/CR-6962 [Chu 2008a].

Quantification of the Frequency of an Initiating Event

It is a commonly accepted PRA practice to estimate initiating event frequencies using operating experience. In case of loss of feedwater transients, a two-stage Bayesian analysis can be used. However, in order to perform such an analysis to consider the contribution of the DFWCS, it is required that the data across multiple plant and vendor designs with varying configurations be collected. Such information is not available in the public domain. As an alternative, in this study, an approach that models a digital system at the level of detail where generic component failure modes and failure data are available was developed. This approach should be applicable to modeling both digital control and protection systems.

An initiating event frequency, f, is the expected number of system failures per unit time. It is related to the reliability of the system $R(T)$, i.e., the probability that the system is operating successfully in time period (0, T), by

$$f = -\frac{\ln[R(T)]}{T} \tag{1-1}$$

Equation (1-1) was derived in NUREG/CR-6962 [Chu 2008a] assuming the initiating event follows a Poisson process with a constant rate, and can be used to evaluate the initiating event frequency, using the $R(T)$ assessed over a time period T, employing a Markov model of the DFWCS. The frequency f is, therefore, the average frequency over the time period T. Note that the actual failure rate of the Markov model changes with time and the use of the average frequency is an approximation. The equation is applicable to any reliability model that calculates a system reliability, including those models that allow component level repair and replacement.

FMEA and Simulation Model

A team of analysts manually undertook the FMEA documented in NUREG/CR-6962 [Chu 2008a]. In many situations, the response of the system to specific individual failures was difficult to

determine, mainly due to lack of documentation and the complex logic modeled in the software. However, even with more complete documentation, it is not feasible to manually determine the system response to multiple failures. Hence, in this study, a simulation tool was developed to facilitate determining the effects of postulated failures. The simulation tool incorporates the software of the CPUs and controllers, and implements rules for assessing whether a loss of automatic control occurs. The tool allows the failures and failure combinations of the components to be postulated, and then represented in terms of their impact on the input and output signals of the CPUs and controllers and associated internal variables of the software. The tool also determines whether a system failure takes place based on the internal states of the system. For example, a detected loss of a steam-generator-level sensor causing a failover, followed by a spurious signal of the watchdog timer associated with the backup CPU, will cause the main feedwater valve (MFV) and feedwater pump (FWP) controllers to switch to the manual mode, which constitutes a system failure. The simulation tool was used to systematically perform the FMEA of all modules of the DFWCS, using the approach described in NUREG/CR-6962 [Chu 2008a] for the main CPU module. It first was employed in identifying those individual failures that directly lead to a system failure, e.g., a failure of the microprocessor of the MFV controller. Such failures are single failures of the system. For those individual failures that do not fail the system, i.e., latent failures, combinations of two failures were considered to identify those sequences of double failures that lead to a system failure. Continuing this process generates higher and higher order sequences. Chapters 3 and 4 discuss the details of the FMEA and simulation tool.

The use of a simulation tool in performing the FMEA can be considered a supporting analysis which plays an important role in developing the reliability model of the DFWCS, just like thermal-hydraulic analyses are used to determine the success criteria and accident timing used in developing accident sequences of a PRA. It is especially important to digital systems due to the complexity of these systems and their use of software. The important role of the simulation tool in determining system success or failure reduces the Markov and ET/FT methods to potential methods solely for quantifying system reliability (i.e., the Markov and ET/FT methods are not used to identify the system failure paths, they are only used to quantify them). Because use of the simulation tool revealed that the order of component failures can be important, the Markov method was selected for quantifying the system failure paths, since it can explicitly account for the order of the failures by defining different sequences of transitions/failures leading to different system states.

Development and Quantification of a Markov Model of DFWCS

A Markov model of a system typically can be represented in terms of a transition diagram showing all the system states and possible transitions among them. It can also be expressed in terms of a set of linear differential equations modeling the transitions among system states, i.e.,

$$\frac{d\underline{P}}{dt} = \underline{M}\,\underline{P}\,, \qquad (1\text{-}2)$$

where $\underline{P}$ represents the probabilities of the system states, and $\underline{M}$ is the transition matrix containing the constant transition rates among the system states. The solution of Equation (1-2) gives probabilistic information about the system. For example, the sum of the probabilities of success states is the reliability, from which the frequency of the initiating event can be calculated using Equation (1-1).

As often is the case for a Markov model, the DFWCS is assumed to initially be in an operable state (i.e., at time = 0). Every time a component of the DFWCS fails, the system transits into another state. An important feature of the Markov model of the system is that components can have different failure modes that entail different impacts on the system. In formulating the DFWCS's transition diagram, all possible transitions in any possible order should be considered. State explosion, i.e., a very large number of possible system states that makes the model too complicated to develop and solve, is a common issue with detailed Markov models. It is addressed in this study by truncating system failure sequences based on their order (i.e., the number of failures in the sequence) and demonstrating that convergence of system failure probability is achieved. This is similar to the concept of cutset truncation (on order) typically done in ET/FT analyses.

The results of the FMEA of the system specify if a system state is a failed state, in which case no additional transitions out of it need to be considered, since the system is already failed. Such a state is called an absorbing state.

For the DFWCS being modeled in this study, a failed component cannot be repaired while the system is operating; therefore, repair does not need to be included in the model. This allows the exact solution of the Markov model to be derived analytically. A simplified solution was developed to compare with the exact solution. Chapter 5 describes the Markov model, and Appendix C contains the detailed derivation of the solution of the model, along with introductory material about Markov modeling solutions. Chapter 7 provides the results of the quantification of the Markov model for the DFWCS.

Data Analysis

NUREG/CR-6962 [Chu 2008a] reviewed publicly available databases for digital system components and performed a Bayesian analysis that attempted to account for the variability of different raw data sources. In the review, potential weaknesses and limitations of the available databases were identified and discussed, and no attempt was made to validate or invalidate the available databases. The limitations in the publicly available failure parameters of digital components identified in NUREG/CR-6962 [Chu 2008a] indicate that additional research and development is needed in this area. This study makes use of the data of NUREG/CR-6962 [Chu 2008a] in developing and quantifying a model of the DFWCS. The data are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes). They are only used in this project to demonstrate the reliability methods and exercise the reliability models.

The data for the quantification were derived from different sources. One important source was the raw data of the PRISM reliability prediction method [RAC PRISM]. The failure rates of many component failure modes were estimated by the HBM [Yue 2006]. For those components whose failure rates were not analyzed in this way, the PRISM RACRate model was used to estimate them [RAC PRISM]. In some cases, the failure rates were taken from other sources, such as NRC-sponsored studies, e.g., NUREG/CR-5500, Volume 10 [Wierman 2002].

In this study, different failure modes for a given component were considered. The failure rates of the different component failure modes were estimated using the failure mode distributions given by Meeldijk [1996] and Aeroflex [2005]. These sources break down the failure rate of a component into its different failure modes.

The failure parameters used in this study have very large uncertainties, and the failure mode distributions are incomplete. Chapter 6 discusses the data analysis and its weaknesses in more detail.

Uncertainty Analysis and Sensitivity Calculations

Parameter uncertainties were propagated in an uncertainty analysis of the top event, with treatment of state-of-knowledge-correlation [Apostolakis 1981]. Also, the effectiveness of several digital-design features, i.e., the redundancy in CPU, use of watchdog timer, and use of demand signal feedback to check for deviations, was explored via sensitivity calculations. Chapter 8 documents these uncertainty and sensitivity analyses. It also discusses modeling assumptions and limitations.

Results and Conclusions

Chapter 9 is an evaluation of the model of the DFWCS against the desirable characteristics of a probabilistic model of a digital system proposed by NUREG/CR-6962 [Chu 2008a]. Chapter 10 provides a high-level, qualitative comparison of the results of this study with those from the studies using dynamic methods [Aldemir 2009]. Chapter 11 discusses the conclusions and insights of the study and summarizes areas where additional research could potentially improve the quality and usefulness of digital system reliability models. It should be pointed out that even though this study models a control system, the approach of this study may also be applicable to protections systems such as a reactor protection system. The conclusions and insights of Chapter 11 are mostly related to modeling methods, and should be applicable to both control and protection systems unless otherwise noted, e.g., the comparison of quantitative results with operating experience of the DFWCS.

2. SYSTEM DESCRIPTION AND SCOPE OF MODELING

This study analyzed the digital feedwater control system (DFWCS) of a secondary loop of an operating pressurized water reactor (PWR). The PWR has two secondary loops, each with a DFWCS. Since the two DFWCSs are symmetrical and do not have many interactions between them, only one of them was analyzed. Also, since complete system design and operation information was not available from the plant, a number of assumptions were made in this study such that the analyzed design should not be thought of as being representative of any particular plant or existing system and may, in fact, include some design features that do not currently exist in any actual DFWCS.

The DFWCS analyzed in this study was described in detail in NUREG/CR-6962 [Chu 2008a]. Here, a summary description of the system is provided, affording readers the information needed to understand the scope of the modeling and the method used in developing the probabilistic model. The loss of automatic control of a DFWCS, given that the plant is in full-power operation during one year, is the top event selected for this proof-of-concept study.

The DFWCS consists of sensors, transmitters, two central processing unit (CPU) modules (the main and backup CPUs), four controller modules (one each for the main feedwater valve (MFV), bypass feedwater valve (BFV), feedwater pump (FWP), and pressure differential indicating (PDI), and associated support systems, i.e., direct current (DC) power supplies and 120v alternating current (AC) buses. The DFWCS sends demand signals to the positioners of the main feedwater-regulating valve (MFRV) and the bypass feedwater-regulating valve (BFRV), and to the turbine controller of the main feedwater pump (MFP). The positioners convert electrical signals into pneumatic pressure that is used to position valves. The PDI controller that normally displays the differential pressure across the MFRV also can serve as a manual control station for the MFRV and BFRV. The digital parts of the system are the CPU modules and controller modules. Each module consists of a microprocessor and its associated components, e.g., analog/digital (A/D) converter, multiplexer (MUX), and digital/analog converter. Figure 2-1, a simplified diagram of the system, shows the modules and components considered in the reliability model of the DFWCS and the main signals between them. The solid boxes represent modules and components that are modeled in detail, while the dotted boxes represent those that are either modeled in a simplistic way or not modeled at all because they are beyond the scope of this study or found not to affect the operation of the system at full power. More discussion of the modeling of the system is provided in the corresponding sections of this chapter.

The system has two modes of operation, automatic and manual. This study assumes that the system is initially operating in automatic mode. The operators can interact with the system by using the controllers that are located in the main control room. If a controller switches from automatic to manual control mode due to a detected failure condition, the operators then can take manual control. This study assumed that a switch to manual mode is a system failure, since automatic control is lost.

The DFWCS also operates in either high-power or low-power mode. Since the plant is assumed to be operating at full power for this study, the system is considered to be initially operating in the high-power mode. During full power operation, the DFWCS normally operates under 3-element control in which the control is based on inputs from three different types of sensors, i.e., steam generator (S/G) level, feedwater flow, and steam flow. These three types of signals are the most important sensor input signals to the DFWCS.

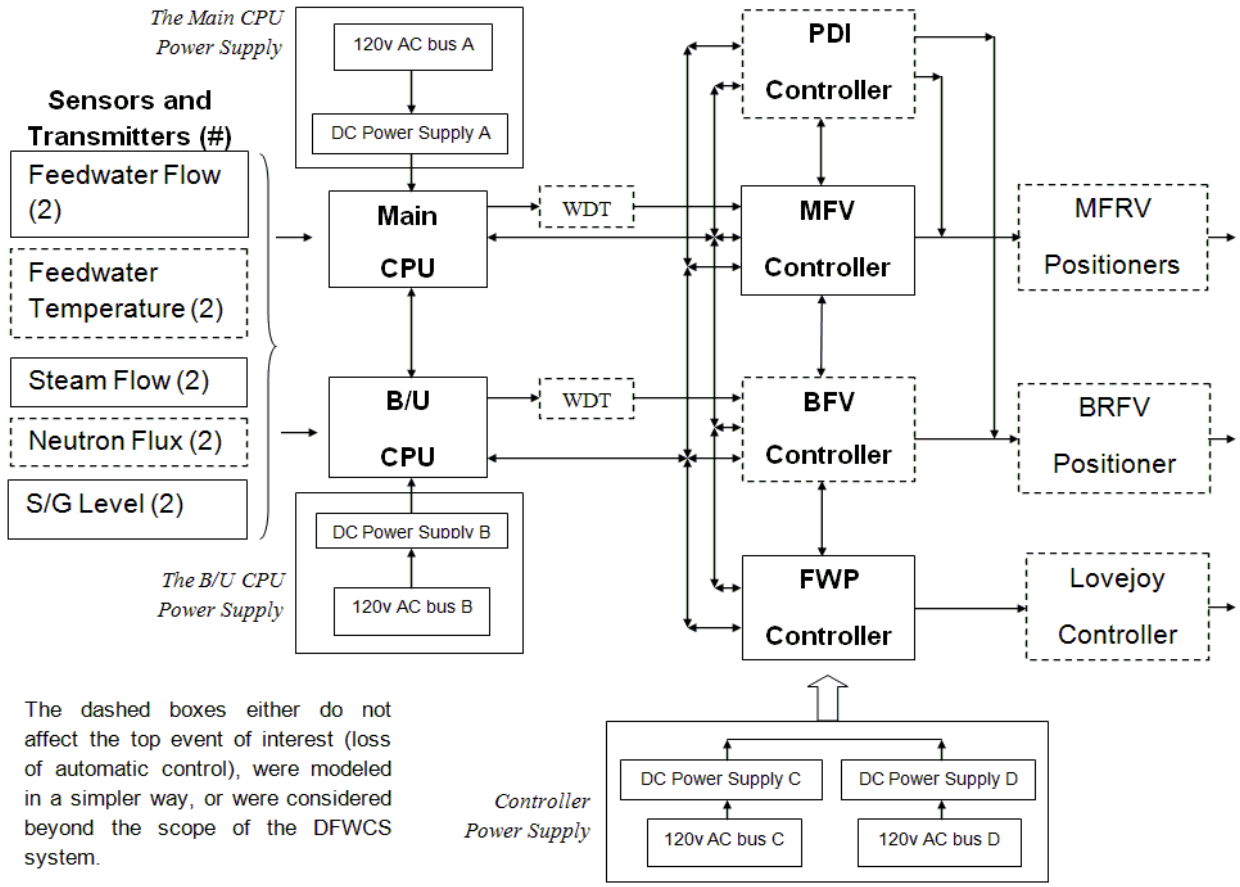

Figure 2-1 Modules of the DFWCS model

The following sections give summary descriptions of the modules and components of the system.

2.1 Main and Backup CPUs and Their External Watchdog Timers

The main and backup CPUs are the brains of the DFWCS. That is, they read the sensor inputs, implement the control algorithms of the DFWCS, and send demands to the MFRV, MFP, and BFRV through the device controllers, i.e., the MFV, FWP, and BFV controllers. System redundancy is provided by the main and backup CPUs. The main and backup CPUs exchange information, such as CPU status, deviations, and input signal validity. Each CPU has an independent external watchdog timer (WDT) that periodically monitors whether the CPU has stopped functioning, i.e., stopped sending the heartbeat signals to the WDT that, in turn, sends the status of its associated CPU to the controllers. Each controller uses the status information to determine which of the two demand inputs (from main or backup CPU) to send to the component associated with that controller. In this study, the main CPU is assumed to be in control, with the backup CPU operating in tracking mode, i.e., taking the demand outputs from the controllers and using them as its own outputs. The tracking mode provides for a smooth transition of control from the main CPU to the backup CPU when the former is determined to have failed, e.g., when the WDT associated with the main CPU detects that the main CPU has

failed. In this study, the WDTs are modeled in a simplistic way. That is, the functions of the WDTs are modeled (e.g., identification of the WDT-detectable failures), while the failure modes of the WDTs, which could be either a failure to indicate the failure status of the associated CPU when the CPU has failed or a spurious signal output indicating that the CPU has failed when it has not, are not modeled due to a lack of design information of the WDTs.

Figure 2-2 is a high-level flow chart of the application software of the main CPU. The main CPU application software includes deviation logic that monitors redundant input signals for possible failures and takes appropriate actions, including notifying the control room operator. The deviation logic for the input signals of S/G level, feedwater flow and steam flow is similar but not identical. In general, the deviation logic consists primarily of sanity checks including out-of-range (OOR) checks, high rate-of-change checks, and deviation checks on redundant input signals. A CPU first determines the validity of certain input signals by checking if the signals are OOR or are changing at a high rate. Depending on the outcome of the validity check, different actions are taken. In the case of:

1. One invalid signal: If the invalid signal is due to failure of a sensor or transmitter, the main CPU ignores the identified invalid signal and uses the remaining good signal in its control algorithm. If the invalid signal is due to internal component failure of the main CPU module and the backup CPU is healthy, then the main CPU will fail itself and the backup CPU will take over control, i.e., a failover to the backup CPU will occur. The determination of whether the invalid signal is due to failure of a sensor or transmitter or failure of an internal component of the main CPU module is based on signal status information exchanged between the main and backup CPUs.

2. No invalid signal: If both signals of the same type are valid, the CPUs will compare them to determine if they differ/deviate significantly. In case of a large deviation of S/G level signals detected by the main CPU, a failover will take place if the backup CPU is healthy. If there is a large deviation between the redundant signals for the feedwater flow or steam flow, the DFWCS automatically switches to 1-element control, i.e., using the signals on S/G level only. This mode of operation is also considered a successful automatic control in this study.

3. Two invalid signals: If both S/G level signals are invalid, a failover will take place provided the backup CPU is healthy. If both signals for the feedwater flow or steam flow are found to be invalid, the DFWCS automatically switches to 1-element control.

In a summary, actions to be taken by the main CPU are determined by (1) types of faulty signals, i.e., level or flow signals, (2) number of invalid signals, and (3) causes of the faulty signals (i.e., due to the sensor/transmitter failures or internal CPU failures). Different actions of the main CPU may vary the system responses to signal failures.

Taking the S/G levels as an example, if a drifting signal due to a level sensor (or transmitter) failure is OOR, the signal will be detected by the OOR check of the deviation logic and the automatic control of the system can be maintained by using the remaining good level signal. The drifting sensor (or transmitter) signal does not have to be OOR, i.e., the signal is still valid but will cause a deviation between redundant level signals. A large deviation will be detected by the deviation check of the deviation logic and cause a failover to the backup CPU. The system will lose the automatic control because the backup CPU detects the same large deviation between the two sensor signals and will also be failed. Note, a small deviation can be coped with by the system.

If a drifting level signal is caused by failure of an internal component of the main CPU, a failover to the backup CPU will occur (the signal either drifts OOR or produces a large deviation between redundant signals) because the backup CPU does not see any problem with the level signals. The automatic control of the system can be maintained.

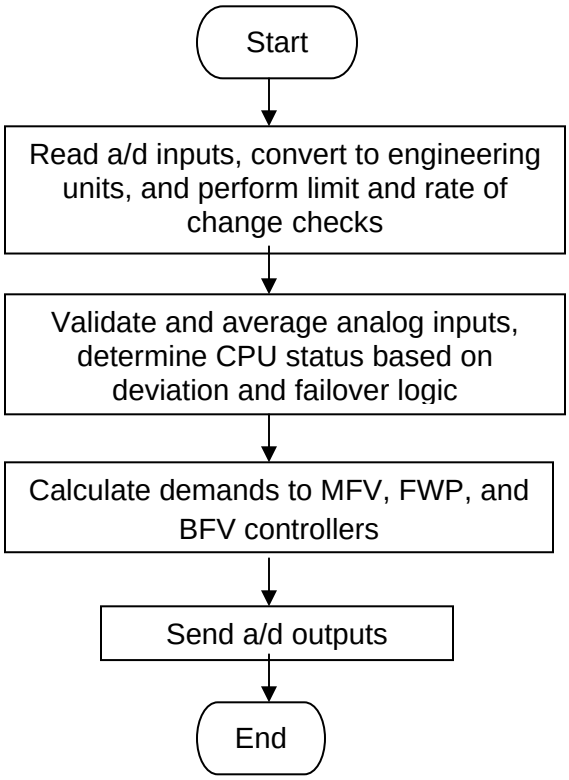

Figure 2-2 High-level flow chart of application software of each CPU

The deviation logic for both feedwater flow and steam flow signals are the same and feedwater flow signals is used as another example. If a drifting feedwater flow signal due to a sensor (or transmitter) failure is OOR, the signal will be detected by the OOR check of the deviation logic and the automatic control of the system can be maintained by using the remaining good feedwater flow signal. If drifting sensor (or transmitter) signal is still valid, it will cause a deviation between redundant feedwater flow signals. A large deviation will be detected by the deviation check of the deviation logic. Because the backup CPU detects the same deviation, the feedwater flow signals are no longer considered usable. However, the system will switch to a 1-element control by using the level signals only and the automatic control (1-element) of the system can still be maintained.

Similar to the case of level signal, if a drifting feedwater flow signal is due to internal failures of the main CPU, a failover to the backup CPU will occur. The automatic control (3-element) of the system can be maintained.

Note, switching to the 1-element control may occur at the same time with an initiating event (e.g., a reactor trip) if the cause is a physical process that is out of control, e.g., too much steam flow. Such an initiating event should be studied separately since it is not due to failure of the DFWCS system.

The CPU application software also has deviation logic on the feedback signals of controller demand outputs. The CPU that is in control does not have OOR checks on the feedback signals, but compares the MFV and FWP demand signals it calculates against the demand signals sent out by the controllers to their associated components; it will fail itself should the feedback demand signal from a controller differ significantly from the calculated demand, i.e., if the main CPU has a large deviation, then a failover to the backup CPU will take place.

2.2 MFV Controller

The MFV controller acts as an interface between the main and backup CPUs, and the MFRV positioners. It also is a manual control station for the MFRV, i.e., the operators may take manual control of the MFRV using it. The CPUs provide valve-position demand signals to the MFV controller that, in turn, relays a demand signal to the MFRV positioners. Normally, the main CPU is in control and the MFV controller sends the demand from this CPU to the two MFRV positioners, PDI controller, CPUs, and CPUs of the other S/G. The MFV controller receives the status of the CPUs from both the CPUs themselves and their associated WDTs. If the main CPU fails and the MFV controller detects it, the MFV controller then uses the demand from the backup CPU as its output. It also sends its automatic/manual (A/M) status to the CPUs, i.e., whether the controller is operating in automatic or manual mode. There is a pushbutton control on the MFV controller allowing the operator to change the S/G-level setpoint manually; this controller can also display the S/G level.

The MFV controller cannot detect its own internal failures, so it cannot prevent the effects of the failures. It has a built-in WDT that may detect certain failures, but will only generate a flashing display on the screen of the controller to alert the operators in the main control room; it does not activate any automatic actions to mitigate the failures. If the MFRV demand output of the MFV controller falls to zero, it will be detected by the PDI controller which then functions as the controller of the MFRV in manual control mode. When any controller switches from automatic to manual control, the system changes its mode of operation from automatic to manual. Therefore, the automatic control function of the DFWCS is lost.

If a failure causes the S/G-level setpoint to deviate, the CPUs will detect the deviation and revert to a built-in setpoint, i.e., the failure is automatically corrected. It was assumed that any failure affecting the display of the level will not affect the function of the MFV controller, and hence, does not have to be included in the model. Figure 2-3 depicts a high-level flowchart of the application software logic of the MFV controller.

The controller clamps the input analog signals within their ranges (e.g., forces an OOR high signal to the maximum value), compares the demand signals from the CPUs to find out if, over a specified duration, they differ by more than a predefined threshold, and generates a deviation alarm when such a deviation is detected. It also transmits the status of the main and backup CPUs back to the CPUs. If the backup CPU detects the failure of the main CPU, then it switches from the tracking mode to the control mode by sending as output its calculated valve demand rather than the demand it receives from the MFV controller output. If both CPUs are found to be failed by any controller, the controller will switch from automatic to manual control, and the system changes its mode of operation from automatic to manual. The MFV controller enters the manual mode by sending the last good demand signal as the output; thereafter, the operators can use the push buttons in the controller to manually control feedwater by increasing

or decreasing the output. In this case, the DFWCS is considered failed because automatic control is lost.

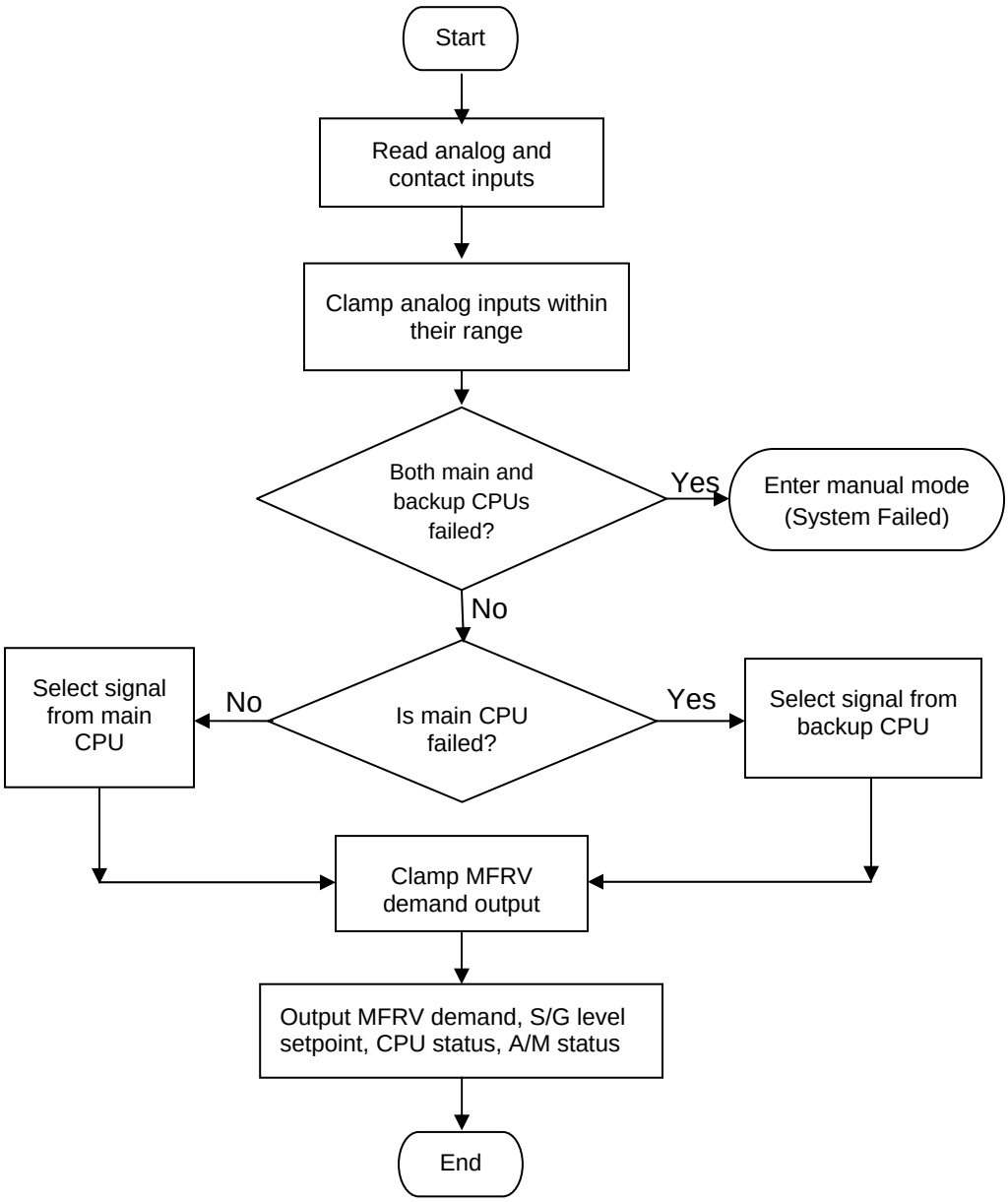

Figure 2-3 High-level flowchart of MFV controller software

2.3 FWP Controller

The FWP controller processes the FWP demand signal in the same way as the MFV controller processes the MFRV demand signal; it receives pump demands and CPU status information from the CPUs and sends FWP demand to the turbine speed controller. Two important differences between the FWP and MFV controllers are that the FWP controller does not send CPU status information to the CPUs, and does not have the PDI controller as a manual backup. Also, the FWP controller has an analog input from the FWP speed-bias potential meter mounted on the main control board. The bias is added to the feedwater pump demand by the CPUs.

This control typically is used to adjust the fraction of the feedwater flow through each of the two pumps, as when starting or securing the second MFP, or matching the CPU's output with that of the manual FWP controller before switching the controller from manual to automatic. The FWP controller monitors the rate of change of the bias signal. If that rate should exceed a preset limit, the FWP controller switches to manual mode, and a bias failure signal is sent to the BFV controller via Microlink[(2)], a network connecting the controllers.

2.4 BFV Controller

The BFV controller processes the BFV demand signal in exactly the same way as the MFV controller processes the MFRV demand signal, i.e., it receives BFV demands and CPU status information from the CPUs and sends the BFRV demand to the BFRV controller. Two important differences with the MFV controller are that the BFV controller does not send CPU status information back to the CPUs, and it does not have the PDI controller as an automatic manual backup. However, the PDI controller can become a manual backup when the operators actuate a control switch. Additionally, the BFV controller provides alarms to the plant's annunciator system and the plant computer, based on failure information received from the MFV and FWP controllers through Microlink. This information includes status of deviation alarms from the CPUs, and status of the "health" of each CPU, i.e., healthy or failed. During full-power operation of the plant, as is assumed in this study, the BFRV is normally closed; even if it fails open, the DFWCS is assumed to accommodate the failure. Therefore, it is not necessary to include the BFV controller in the reliability model except possibly to account for its failure modes that can logically affect the system operation, that is, the failure modes associated with its A/M status. As discussed in Section 3.3.5, an explicit BFV controller model is not necessary because the failure of the A/M status can be included in the FMEA for the main CPU.

2.5 PDI Controller

Although the PDI controller is connected to both the MFV and the BFV controllers, it normally is on standby and does not directly undertake any control function during DFWCS automatic control. It normally displays the differential pressure across the MFRV, and has a buffer for holding the outputs of the MFV and BFV controllers until the PDI controller is automatically or manually switched into the control loop.

The MFV demand output is also sent to the PDI controller. The sum of the MFV demand output and the PDI demand output is the demand that will be sent to the MFRV. During normal operation, the MFV demand output is not zero and the PDI controller output is zero (implemented by the software of the PDI controller). If the PDI controller detects that the MFV demand output fails to zero, then the PDI controller will raise its output to the pre-failure MFV demand value, which will be added to the actual MFV demand output to support manual control of the MFRV.

Because the PDI supports manual control of the MFRV after the demand from the MFV controller falls to zero, failures of the PDI do not affect the likelihood of the initiating event, i.e., loss of automatic control. An exception is the failure mode of the PDI wherein it incorrectly takes over control of the MFRV from the MFV controller, thus causing a loss of automatic control because control then becomes manual. As discussed in Section 3.3.5, an explicit

[(2)] Section 4.5.1 of NUREG/CR-6962 [Chu 2008a] provides a detailed description of Microlink. A loss of the Microlink communication network affects alarm and time synchronization only, and does not affect control since CPUs and device controllers are asynchronously running. Therefore, it is excluded from this study.

PDI controller model is not necessary because the false takeover by this controller can be included in the FMEA for the MFV controller.

If the BFV controller fails such that its analog demand output drops to zero, a manual hand switch can be actuated so that the PDI controller can be used to manually control the BFRV.

2.6 Other Components

Sensors – Sensors provide analog signals through the transmitters to the CPUs. There are five different types of signals, viz., feedwater temperature, feedwater flow, steam flow, neutron flux, and S/G level. Each type of signal has two sensors and transmitters. A CPU module converts the analog signals into digital signals via a MUX and an A/D converter. As discussed in Section 2.1, the CPU software checks the validity of the digitized input signals (OOR, bypass, and rate of change), and any deviation between two signals of the same type. Different deviation and validity logic processes each different signal type. Since feedwater temperature signals are not used during the high-power mode of operation, they are not modeled. Similarly, loss of the neutron flux sensors during this mode does not need to be modeled because it only inhibits a transfer to low-power mode. Chapter 3 discusses the sensor failure modes.

MFRV Positioners – The positioners are microprocessor-based current-to-pneumatic (I/P) devices that convert the input current signal from the MFV controller to a pressure signal, which positions the valve. In general, the generic FMEA model of a digital module discussed in Chapter 3 is applicable to the positioners. In this proof-of-concept study, the positioners were not modeled because sufficient information on their design and operation was not available.

BFRV Positioner – The positioner is a microprocessor-based I/P device that converts the input current signal from the BFV controller to a pressure signal, which positions the valve. For the same reason that there is no need to model the BFV controller because its failure is expected to be accommodated by the DFWCS, the BFRV positioner also does not need to be modeled.

Turbine Controller – The turbine controller is a digital controller, receiving demand signals from the FWP controller and controlling the FWP accordingly. In this study, the turbine controller was not modeled because sufficient information on its design and operation was not available.

DC Power Supplies – Each CPU has its own DC power supply with its own 120v AC bus. The four controllers are assumed to share two DC supplies, each fed by a different 120v AC bus.

120v AC Buses – It is assumed that four different 120v AC buses supply power to the DC power supplies of the CPUs and controllers, two for the CPUs and two for the controllers.

Heating, Ventilation, and Air Conditioning (HVAC) – In general, digital systems may fail if exposed to elevated temperature. However, the dependency on HVAC was not considered to be significant because the DFWCS is located in the control room, and the effect of a loss of HVAC would take hours to develop and can be easily recognized and mitigated, e.g., by opening a door.

3. IDENTIFICATION OF INDIVIDUAL FAILURE MODES AND THEIR EFFECTS FOR THE DFWCS

This chapter documents the failure modes and effects analysis (FMEA) of the digital feedwater control system (DFWCS) components and associated support systems. An FMEA approach is used that was conceptualized in NUREG/CR-6962 [Chu 2008a], wherein the level of detail is driven by availability of generic component failure data. Reliability Prediction Methods (RPMs), such as those in Military Handbook 217F [Department of Defense 1995], PRISM [Reliability Analysis Center (RAC) manual], and Telcordia [2001], are the only publicly available databases for digital components identified for this study. Their weaknesses are that the estimates of failure rate may be inaccurate due to use of conservative assumptions and lack of applicable data [Gu 2007, Pecht 1994], and uncertainties are not considered. Further, they use parts count and part stress methods that are applicable only to systems without redundancy. NUREG/CR-6962 [Chu 2008a] provides more discussion of the RPMs. The FMEA approach of this study is at the level of detail of the data in the PRISM database, but also considers the different failure modes of the components. It can support the development of more realistic models of digital systems.

A digital system is envisioned as consisting of modules, each comprising common generic components, such as an analog/digital (A/D) converter, a multiplexer (MUX), a microprocessor and its associated components (e.g., random access memory (RAM) and buses), a demultiplexer (DEMUX), and a A/D converter. By considering the generic components, the approach can be applied to any digital module of digital systems. In the FMEA, generic failure modes of the components are postulated in terms of the output signals associated with the components, and then the effects of the presupposed failures are determined by examining how the rest of the system processes the signals. Chapter 6 describes the usage of the raw data of the PRISM database, along with other sources of failure parameters. The failure parameters used in this study are generic, estimated using available data. In general, data that is specific to the component type, application, and operating environment is more appropriate, but often unavailable. The parameters in this report are presented for illustrative purposes only, i.e., to illustrate the approach and methodology of this study. The parameter values presented are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).

This chapter details the FMEA approach using the main central processing unit (CPU) module and the feedwater pump (FWP) controller module as examples. Appendix A gives complete FMEA tables of the system. Initially, the FMEA was carried out manually by reading various documents about the system, i.e., the system description, requirement specifications, hazard analyses, pseudo software (i.e., high level description of software using the software programming structure, but with the program details not included), and piping and instrumentation diagrams. Soon thereafter, a simulation tool was developed to automate the process due to difficulties in manually relating different pieces of information in determining the effects of individual postulated failures. This tool, discussed in Chapter 4, was based on the actual source code for the system and was employed to verify the results of the manual FMEA for individual failures. In turn, the manual FMEA provided a check for the simulation tool. More importantly, the simulation tool was the only practical way to assess the effects of combinations of postulated failures and their sequences. The FMEA, including the simulation tool, is a supporting analysis which plays an important role in developing the reliability model of the DFWCS, just like thermal-hydraulic analyses are used to determine the success criteria and

accident timing used in developing accident sequences of a PRA. The simulation tool is especially important to digital systems due to the complexity of these systems and their use of software.

3.1 General Issues with Current FMEAs for Digital Systems

FMEA is a method used to identify the failure modes of components of a system and their subsequent effects on the system. FMEAs, which usually are conducted at different levels of detail, can be used to support development of system reliability models. The highest level of detail is the entire system. The system can then be decomposed and FMEAs conducted at lower levels, e.g., subsystem levels. The FMEA at a particular level is used to start the next lower level FMEA since the failure modes of one level indicate the effects of failure at its immediate lower level. The process of decomposition may continue until the available information cannot support a more detailed analysis, or the purpose of the FMEA does not require more detail.

While these discussions are applicable to all system FMEAs, existing generic issues with digital-system FMEAs are (1) there is no well-established definition of the failure modes and their effects for digital systems and (2) there is no specific guidance of how to undertake FMEAs for digital systems. Despite these existing issues several reliability studies of digital systems have been completed, e.g., those discussed in Chapter 8 of NUREG/CR-6962 [Chu 2008a]. In general, those studies were not conducted with sufficient detail for the approach described here; i.e., the failure modes of a component either were not explicitly defined or often were implied as "failures to perform its dedicated function," so that the only identified effect of failure on the system is that the system has failed.

Current digital systems are highly complicated. Theoretically, all the relevant interactions between the components of a digital system should be captured by its reliability model. In practice, these interactions are hard to capture without using appropriate FMEAs at proper levels. Lacking a quality FMEA, it is difficult to create high-fidelity reliability models.

In nuclear power plants, digital systems mainly are employed to control specific equipment or a process, or perform safety-related functions, such as tripping the reactor or actuating an emergency safety feature. Therefore, differences in desired functions and the uniqueness of individual industrial processes require specific digital system design features. This implies that the data for a specific digital system of interest generally are very scarce. The collecting of suitable data or system-specific data is exacerbated by the fact that digital systems are likely to be upgraded frequently.

Although the designs of digital systems can be very different from each other, usually they all use generic digital components, e.g., microprocessors and A/D converters. As long as the impacts of the failure modes of these generic components on the associated signals are clear, these component failure modes are suitable for developing a reliability model based on the system design/configuration information. If specific data are unavailable, then generic failure data for these components will have to be used in the model.

Although there is not a standard list of failure modes for digital components, in general, the failure modes of a generic component can be defined in terms of its function(s). Therefore, a consistent set of failure modes can be applied to components of the same type, even if they are

of different makes or models. Defining failure modes in such a way allows them to be used in reliability analysis as long as the associated failure data are available.

3.2. A Generic Approach to the FMEAs of Digital Systems

As discussed previously, it is preferable to carry out an FMEA at the level of generic components where reliability failure parameters are available because then the reliability model developed using the FMEA can be quantified using these failure parameters. Another advantage of performing FMEAs of digital systems at that level comes from the fact that software runs on a microprocessor. The software and hardware interaction, the fault-tolerance characterized by specific software design, and the interactions between digital systems and monitored/controlled processes are reflected by signals transmitted between generic components of the digital system; therefore, they can be potentially captured by the FMEA at this level. For example, a specific failure mode of a MUX may generate a signal failure that the fault-tolerance design will detect and correct without failing the entire system.

Accordingly, a generic approach to undertaking FMEAs of digital systems is proposed here. The entire digital system is decomposed into different levels of detail until the level of the generic components is reached. The number of intermediate levels depends on the complexity of, and information available about, the particular system. Failure modes are postulated at the lowest levels (in this case, at the level of generic components), and their effects propagated to higher levels until the impacts on the entire system can be determined.

Figure 3-1 illustrates the FMEA process adopted in this study and applied to the DFWCS. The key points in performing FMEAs are that (1) the status of the system eventually is determined by module signals that reflect the interactions between the modules and between the digital system and the plant and (2) these signals are directly affected by the status of the generic components of the modules. Figure 3-1 depicts a pathway showing how the failure modes of generic components in a specific DFWCS module affect its status and that of the whole system. That is, the failure modes of generic components are used to evaluate their impacts on the module input and output (I/O) signals, which in turn determine the status of the entire system, i.e., whether or not a system failure takes place .

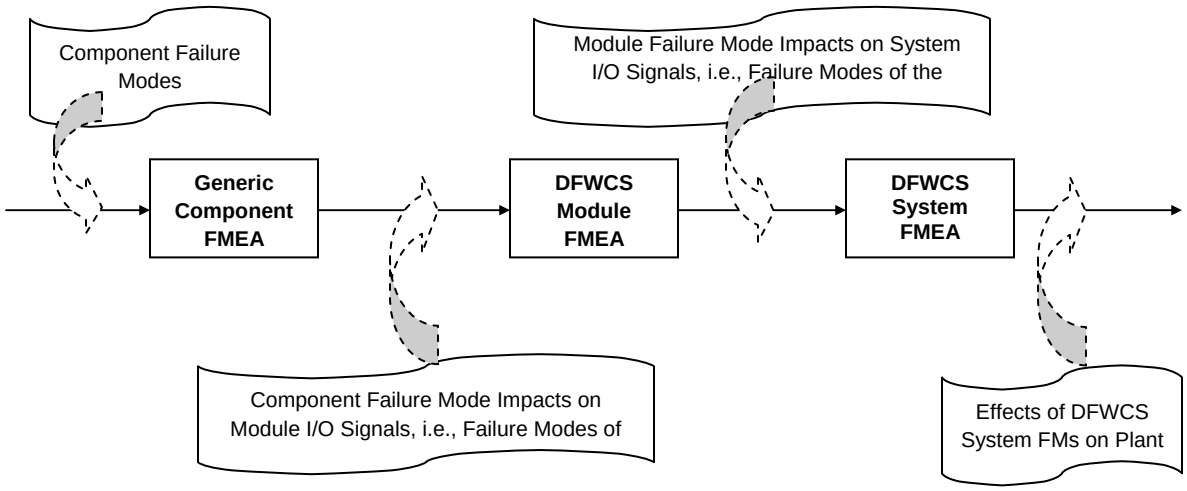

Figure 3-1 Steps in the generic FMEA approach applied to the DFWCS

The DFWCS consists of: (1) the main and backup CPUs that essentially execute identical software and use the same control algorithms to calculate control demands based on input from the plant and (2) four controllers that receive the corresponding demand signals from the CPUs and forward them to valve positioners or the pump speed controller. The interactions of the two CPU modules and four controller modules are determined from system design information. Each DFWCS module can be considered as a complete digital system with its own A/D input, processing, and A/D output. Therefore, the major components of all modules include A/D converters, A/D converters, and microprocessors and their associated peripheral devices (e.g., RAM, read-only memory (ROM), and buses), MUXs, DEMUXs, and some analog I/O devices (e.g., current loop devices). The major components of the modules are identified based on general architecture of digital systems.

Effectively, the DFWCS is broken down into three levels. The highest level is the entire DFWCS system, and the lowest level corresponds to the generic digital components. The single intermediate level (the module level) includes the six modules.

The failure modes at the system level are the failure effects of the modules; similarly, the failure modes at the module level are the failure effects of the components within the module. It should be noted that system failure modes usually are defined in terms of the system's functionalities and thus, are system specific. The FMEA's scope encompasses the internal failures of the system, but excludes external events, such as fire or seismic events.

System-Level FMEA:

> For the system-level (top-level) FMEA, the scope of analysis included the entire DFWCS. As described in Chapter 2, system failure is defined as the loss of automatic control while the plant is operating at full power.

Module-Level FMEA:

> The next level of the FMEA included the major modules of the DFWCS, i.e., the main CPU, backup CPU, MFV controller, BFV controller, FWP controller, pressure differential indicating (PDI) controller, and some related dependencies, such as power supply and sensors. The failure modes of these major modules are represented by the failures of their individual I/O signals (see more discussion in Section 3.3.1); their impacts on the behavior of the modules were analyzed. It is noted that the reliability model of this study does not include the BFV controller and the PDI controller. The reason for excluding them is discussed in Sections 2.4 and 2.5, as well as in Section 3.3.5.

Major-Component-of-Module-Level FMEA:

> The lowest level FMEA analyzed the components inside the modules of the DFWCS. The controllers are application-specific integrated circuit (ASIC)-based devices. Since the major components of both controllers and CPUs are similar, they are analyzed in the same way.

FMEAs at different levels are performed either by a "bottom-up" or a "top-down" approach. In fact, the former approach is preferred and was adopted for the DFWCS because of difficulties in deductively identifying all possible causes of a given failure, as discussed in Chapter 4.

3.3 FMEAs of the DFWCS Using the Generic Approach

In this study, steady-state operation of the system is assumed as the initial condition, and loss of automatic control is the system failure condition being modeled. The following are important assumptions made in performing the FMEAs:

- All components, including those playing a standby role, e.g., the backup CPU, are operating at all times and can fail at any time.

- Typically, a component can have more than one failure mode with different effects that must be modeled differently. A component is assumed to fail only once in a given failure sequence, i.e., after one failure mode of the component has occurred, other modes cannot occur for the same component. This assumption is believed to hold for most of the digital components, because available information on digital component failures seems to suggest so, i.e., the hardware failure databases reviewed in NUREG/CR-6962 [Chu 2008a] did not provide any indication that additional failures may occur subsequent to an initial failure. It would be unrealistic to assume that a component can always fail more than once. It may be possible that a certain component fails to an intermediate failure mode before it reaches one of the other failure modes. If recognized, such a sequence of failures can still be analyzed and modeled using the approach of this study as discussed in Section 4.2.7.

- Due to lack of detailed design information, failures of different components are assumed to be independent of each other (regardless of how they are physically wired together). It is recognized NUREG/CR-6962 [Chu 2008a] that determining the effects of component failure modes in a real digital system could be much more complex than what this study assumes. For example, the detailed connection of a digital output to a few digital inputs determines if failure of one input would affect other inputs, which suggests that cascading component failures may occur. On the other hand, built-in mechanisms that may detect and isolate the cascading faults can also be designed, and included in evaluation of FMEAs as needed. The independence assumption is introduced because, otherwise, detailed analyses of the designs at the circuit level, which are unavailable in this study, must be performed for individual components to determine how a specific failure of a component affects the connected components.

- It is assumed that a drifting signal will eventually drift to out-of-range (OOR) high or low. As a result, in the model a drifting signal is always detected by the OOR check, and the system may continue to successfully operate, e.g., using the redundant signal (i.e., the signal that does not drift OOR). This treatment may be non-conservative because, in reality, a drifting signal may not drift OOR, and may cause an undetectable failure that could result in system failure. However, as discussed in Section 2.1, if a drifting signal is not detected by the OOR check, it still may be detected by the built-in deviation check of the application software. Section 8.4.4 provides more discussion on the significance of the assumption used in this study regarding signals drifting OOR.

- Ideally, for a control system, a thermal-hydraulic model of the plant would capture a drifting signal. On the other hand, recognizing that such a failure mode may cause system failure, the failure mode can be modeled accordingly. The only difficulty may be

estimating its failure rate. More discussion on how to better model drifting signals is provided in Section 4.4.

- If the failure effects of a component failure mode are unknown due to a lack of knowledge about that failure mode, e.g., loss of a basic input/output system (BIOS), it is conservatively assumed that the associated module is failed. In addition, if the component failure mode causes the undetectable failure of the main CPU or any of the controllers, the entire DFWCS is assumed to fail. Undetectable failure of the backup CPU does not directly lead to loss of automatic control of the system (i.e., DFWCS failure), but the automatic control will be lost, if there is a need for failover from the main CPU to the backup CPU.

3.3.1 FMEA of the Main CPU Module

This section provides detailed illustrations of how failure modes are defined, and how they are propagated to the system level. Appendix A gives complete FMEA tables of the DFWCS. It is anticipated that FMEAs can be carried out for other digital systems using a similar process to that used for the examples covered in this section.

Figure 3-2 shows the "internal" components of the main CPU module of the DFWCS, i.e., the components connected to the main CPU, and considered in the reliability model as its internal parts.

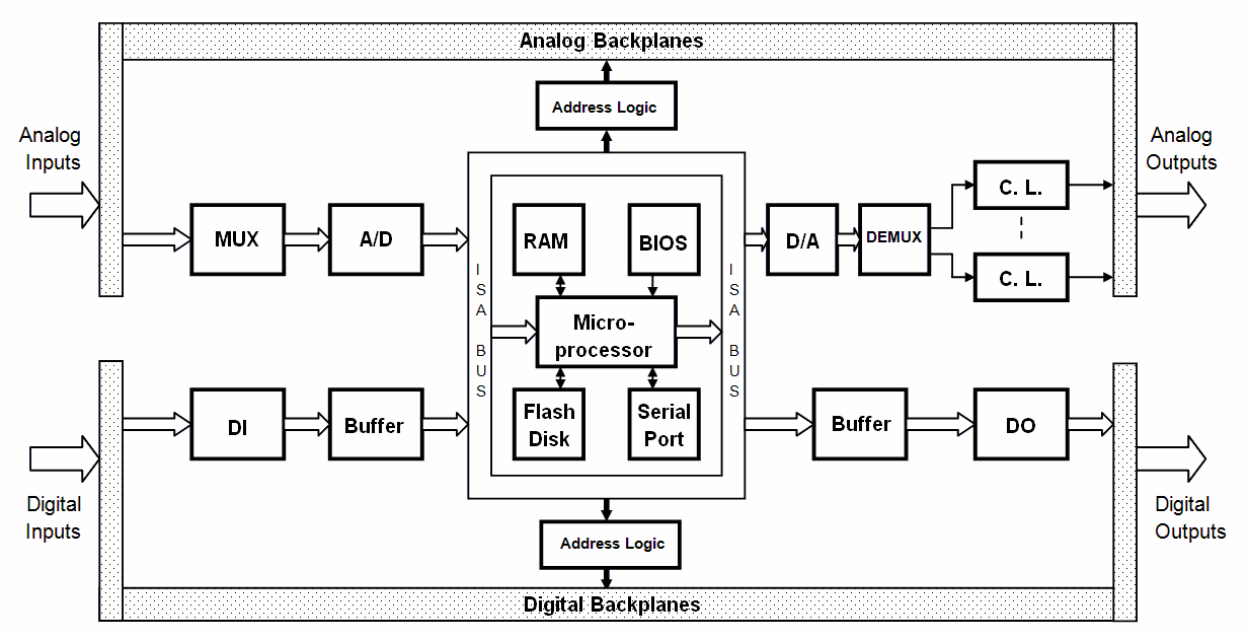

Figure 3-2 Major components of the main CPU module

In the diagram, analog backplanes and digital backplanes are buses that interface with all I/O of the main CPU module (for simplicity, analog backplanes A and B are combined in the figure). An Industry Standard Architecture (ISA) bus is used for the microprocessor of the main CPU module to interact with components connected to the backplanes. A current loop device produces a current output (usually 0-20 milliamperes (mA)). Each analog output is assumed to

use one current loop. The figure does not depict the current loops for analog input signals, but it also is assumed that each analog input uses a current loop. Other components are all standard in digital systems. The arrows represent signal flows between different components.

The failure modes for individual components of the main CPU module are summarized below, and the sources of the failure modes are cited. Chapter 6 discusses the breakdown of component failure rates into their constituent failure modes through the use of failure-mode distributions.

1. Hardware Common-Cause Failure (CCF): The hardware of the main CPU and backup CPU is identical. The occurrence of a hardware CCF may fail the entire system.

2. Software: The main and backup CPUs run the same software and a software CCF may occur and fail the entire system. Two failure modes are considered: (1) the software on the main CPU seems to be running normally but sends erroneous output and (2) the software halts and hence, the CPU stops updating output. In addition to the CCFs of software, the above failure modes are also considered for the individual software running on the CPU modules considering the fact that the main and the backup CPU are in different modes (controlling and tracking modes) and might be running different portions of the software at any given time. More information on the completeness of software failure modes is provided in Section 8.3.

3. Microprocessor of the Main CPU: Failure modes considered are (1) the microprocessor seems to be running normally but sends erroneous output and (2) the microprocessor stops updating output [RAC 1997b].

4. Associated Components of a Microprocessor, such as the ISA bus, RAM, ROM, BIOS, flash disk, buffer, and serial port: It is conservatively assumed that each component has only one failure mode, i.e., a loss of the component, which entails the loss of the functions performed by the component.

5. Address Logic: This is a generic digital component, also called a decoder. A microprocessor uses the address logic to access the information transmitted on the backplanes. The failure mode is assumed as a loss of the address logic, so that the microprocessor cannot access the intended information upon loss of the address logic.

6. Voltage Input Module: The voltage regulators are assumed to be the major component of the voltage input module of the main CPU. The failure modes are fail-high and fail-low of the associated voltage input signal [RAC 1997b].

7. MUX and DEMUX: Failure modes of MUXs and DEMUXs are defined in Aeroflex [2005] in terms of the analog signals they process, which include a loss of one or all signals. No other failure modes of MUXs or DEMUXs were mentioned in Aeroflex [2005], and, therefore, a loss of signal is modeled in this study as signal fails low.

8. A/D and Digital/Analog (D/A) converters: Both A/D and D/A converters are linear integrated circuits (ICs), i.e., the I/Os are proportional to each other; all analog I/Os of the same module share them. The failure modes of an A/D converter include all bits of the A/D stuck at zeros, all bits stuck at ones, and a random bit-failure of the A/D converter [Meeldijk 1996]. The failure modes of a D/A converter include output fails

(drifts) high or low [Meeldijk 1996]. It is assumed that if the D/A converter output starts drifting, it will eventually reach the high or low detection threshold.

9. Current I/O Modules: The major components of the current I/O modules are current loops that essentially are linear transmitters/receivers. They also are linear ICs and their failure modes are current signal fails (drifts) high or low [Meeldijk 1996]. It is assumed that if the current starts drifting, it will eventually reach the high or low detection threshold.

10. Digital I/O Modules: Digital I/O is implemented via a solid-state switch [Eurotherm 2000]. The status of a digital signal is controlled by opening or closing the switch. The solid-state switch may fail to operate (fail as is) and spuriously operate (fails to the opposite state), as stated in RAC [1997b].

In summary, failure modes of components that carry analog signals include “signal fails high” and “signal fails low” (a loss of signal is modeled as signal fails low, as indicated above). Failures of drifting analog signals, such as random signals, are assumed to either drift high or drift low, i.e., the same as fail high or fail low. This assumption about drifting signals will be further discussed in Section 4.4. Failure modes of components that carry digital signals include normally closed, fails closed (NCFC), normally closed, fails open (NCFO), normally open, fails closed (NOFC), and normally open, fails open (NOFO). These failure modes will cause the corresponding digital I/O signals of a module to fail to operate (NCFC and NOFO) or fail to the opposite state (NCFO and NOFC). Impacts of the failure modes of other components on the modules were discussed above and in Table 3-1.

It is noted that the failure modes of components discussed in these references may not perfectly match the component failure modes of the main CPU module of the DFWCS; nevertheless, they were the best approximation found at present. For example, “no output” and “short-circuit” failures modes of a linear IC, such as A/D and D/A converters, are interpreted as “fails low.”

Table 3-1 lists the failure modes of representative generic component types and their potential impact on the main CPU module and the DFWCS. The impacts on the main CPU module and the system were determined by the FMEA performed manually in NUREG/CR-6962 [Chu 2008a] and validated with the automated FMEA tool discussed in Chapter 4. Impacts of some of the failure modes were postulated based on understanding of the function and design of the components, e.g., a loss of BIOS is assumed to be an undetectable failure that will fail the system. The table does not provide a complete FMEA of the main CPU module; instead, there is an explanation of the meaning of the failure modes of generic component types, and an illustration of the way these failure modes propagate to the entire system via the intermediate module level.

It is important to consider fault-tolerance features in each CPU module. If the main CPU module fails and is detected by these features, the backup CPU module will assume control of the system. This process is named a “failover to the backup CPU,” or simply a “failover.” Each CPU module (main and backup) has available two types of fault-tolerance features. The first one is failure-mode detection by the application software running on the CPU, and the second one is monitoring by an external watchdog timer (WDT). Each feature can initiate a failover from the “controlling” CPU module (normally the main) to the “tracking” CPU module (normally the backup).

This study defined a system failure as the loss of automatic control of the DFWCS. Based on the physical meaning of the failure modes of a specific component, their impact on signals associated with this component can be determined. Thus, the effects of component failure on the main CPU module can be established, i.e., the failure modes of the main CPU, based on the system design information. The impact of the main CPU failure modes on the entire system can then also be evaluated based on the system design information.

Column 1 in Table 3-1 presents the failure modes for individual components (including software) in the main CPU module. Column 2 (heading "Failure Mode Detected by") indicates whether the failure modes can be detected by the application software or the external WDT, which represent fault-tolerance features of the main CPU. The impacts of the failure modes on the main CPU module are indicated in Column 3. Column 4 establishes whether a failure mode triggers system failure. Note that failures of different signals carried by a particular type of component, e.g., current loop, may have different impacts on the main CPU or the DFWCS system, i.e., the impact is signal dependent. The actual failure effects for each specific component are provided in the FMEA tables of Appendix A. Finally, Column 5 provides comments on each failure mode.

Considering the failure modes of the main CPU, an undetectable failure will result in failure of the DFWCS (i.e., loss of automatic control) because the main CPU is assumed to be the controlling CPU. A failure mode detectable by the application software indicates that the main CPU can detect the failure, so that the application software initiates a failover to the backup CPU, if needed. A WDT detectable failure signifies the detection of the failure mode by the external WDT of the main CPU, and the resulting failover to the backup CPU. For a failure mode of a component that does not impact the main CPU, e.g., a loss of serial port, the main CPU will continue to carry out the DFWCS control function. If a failure mode does not cause the main CPU module to fail, this module will continue to operate with a latent failure present, i.e., a failure that may subsequently lead to failure of the DFWCS if combined with other component failures.

The failure rates of these major components are required to quantify the digital system reliability once the reliability model is created. Another important parameter is the distribution of the component failure modes. Usually, the failure rate of a digital component includes all its failure modes. Because different failure modes may have different effects, that distribution, i.e., the distribution of the different failure modes of a component with respect to the "total" failure rate of the component, also is needed. Accordingly, the "total" failure rate must be split into the individual failure rates of the component failure modes. This study mainly adopted the failure mode distributions of different components described in Meeldijk [1996] and RAC [1997b]. Chapter 6 discusses in detail the reliability data of digital systems and components. The approach described above for the main CPU module was also employed to analyze other modules of the DFWCS. These analyses are summarized below.

3.3.2 FMEA of the Backup CPU Module

The hardware and the software of the backup CPU module are identical to those of the main CPU module. During plant normal operation, the main CPU module is in the "controlling" mode, i.e., it is controlling the components associated with the DFWCS, such as the main feedwater-regulating valve (MFRV) and FWP, and the backup CPU module is in "tracking" mode.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Software CCF	-	-	-	Yes	It is assumed that the CCFs of software or hardware will fail the entire system. Therefore, detection of the failure is not an issue. Section 6.3 describes how CCFs are modeled in this study.
Hardware CCF	-	-	-	Yes	
The software on the main CPU seems to be running normally but sends erroneous output	No	No	Undetectable Failure	Yes	This is considered an undetectable failure of the main CPU and will fail the entire system.
Software halt (CPU stops updating output)	No	Yes	WDT Detectable Failure	No	When the WDT no longer receives a toggling signal, it will cause a failover of the main CPU to the backup CPU provided that the status of the WDT is normal.
The microprocessor seems to be running normally but sends erroneous output	No	No	Undetectable Failure	Yes	This is considered an undetectable failure of the main CPU and will fail the entire system.
The microprocessor stops updating output	No	Yes	WDT Detectable Failure	No	When the WDT no longer receives a toggling signal, it will cause a failover of the main CPU to the backup CPU provided that the status of the WDT is normal.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Loss of ISA bus	No	Yes	WDT Detectable Failure	No	The input and output of the CPU rely on the ISA bus, and both the application software and the WDT can potentially detect this loss of the ISA bus. However, it is assumed that this CPU failure is detected by the WDT if its status is normal because the application software may be unable to send out any alarm or signal regarding failure of the main CPU due to the loss of both the input and output of the CPU.
Loss of RAM	No	Yes	WDT Detectable Failure	No	Application software has to be loaded into RAM to run it. Thus, the application software cannot run upon a loss of RAM. It is assumed that the WDT can detect the loss of RAM because the software of the main CPU will no longer run and send out toggling signals.
Loss of BIOS	No	No	Undetectable Failure	Yes	The input and output operations of the CPU rely on BIOS routines. However, it is unknown whether a loss of BIOS will cause a complete loss (or a partial loss) of inputs to and outputs from the application software and CPU; hence, the failure is conservatively assumed to be undetectable.
Loss of flash disk	No	No	Undetectable Failure	Yes	The failure effects of a loss of the flash disk that stores software are unknown. The failure is conservatively assumed to be undetectable.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Loss of serial port	Yes	No	Continued Operation	No	Communication between the main CPU and power distribution unit is via a serial port. From plant information, the CPUs send data to the power distribution unit for display and the setpoint can be changed there; the change then is sent to the CPU via the serial port. Apparently setpoints are changed offline. Therefore, a loss of the serial port will not affect main CPU normal operation.
Fail (drift) high or fail (drift) low of current loop device	Signal dependent	No	Signal Dependent	Signal dependent	The current loop is a linear device that may fail high or low, resulting in the associated I/O signal failing high or low. Fail low includes failures of fail to zero. The failure modes of the current loop device cause the associated signal to fail high or low. The main CPU processes different signals differently. For example, failure of level signals will cause the backup CPU to take over control from the main CPU based on the software logic. Further analysis is needed for individual signals to determine their impacts on the main CPU module and/or the DFWCS.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Fail (drift) high or fail (drift) low of voltage signal	Signal dependent	No	Signal Dependent	Signal dependent	The voltage regulator is a major component for the voltage signal I/O. It may fail high or low, and effectively, causes the voltage signals to fail high or low. Again, further analysis of individual signals is needed to determine their impacts on the Main CPU module and/or the DFWCS.
Loss of all signals from MUX	Yes	No	Application Software Detectable Failure	No	Loss of a signal means that the signal fails low. All analog inputs share the MUX. This failure mode indicates that all analog signals related to this MUX fail low.
Loss of one signal from MUX	Signal Dependent	No	Signal Dependent (Application Software Detectable, Undetectable, or Continued Operation (with Latent Failure)	Signal dependent	The failure mode indicates a loss of a specific analog signal. The responses to this failure depend on the specific signals.
Loss of all signals from DEMUX	Yes (but cannot be corrected by the CPUs)	No	Undetectable Failure	Yes	1. The DEMUX is similar to the MUX. It is shared by all analog outputs. Loss of a signal means that the signal fails low. 2. Based on the system design information, this failure will cause a loss of automatic control, which this study defines as a system failure.
Loss of one signal from DEMUX	No	No	Signal Dependent (Undetectable Failure or Continued Operation)	Signal Dependent	1. The failure mode indicates a loss of a specific analog signal. 2. Responses to this failure depend on the individual signals.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
All 16 bits of A/D converter stuck at zeros or ones	Yes	No	Application Software Detectable Failure	No	1. Both A/D and digital/analog converters are linear ICs. The A/D converter is shared by all analog inputs, and its loss will entail the loss of all analog inputs. 2. Stuck at zeros or ones indicates that all analog signals fail low or high. The main CPU software can detect failures of some input signals, and then cause a failover.
Random bit failure of A/D converter	No	No	Undetectable Failure	Yes	Although the main CPU software can detect some random failures, they are conservatively assumed to be undetectable and will fail the whole system.
Output of digital/analog converter fails (drifts) high	Yes	No	Application Software Detectable Failure	No	1. The digital/analog converter is shared by all outputs of the main CPU, and its loss will result in a loss of all outputs. 2. This failure will cause a failover to the backup CPU by the main CPU application software.
Output of digital/analog converter fails (drifts) low	Yes (but cannot be corrected by the CPUs)	No	Undetectable Failure	Yes	This failure will cause a loss of automatic control of the DFWCS, defined in this study as a system failure.
Loss of address logic	No	No	Undetectable Failure	Yes	The address logic also is called a decoder. Although some failures of address logic might be detected by the application software, it is conservatively assumed that a loss of the address logic will result in an undetectable failure of the main CPU and fail the system.

Table 3-1 Illustrative examples of performing FMEA at component level of the Main CPU module.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Loss of output buffer	No	Yes	WDT Detectable Failure	No	All digital I/Os rely on buffers. Loss of the output buffer will cause the main CPU to fail to send out a toggling signal to the WDT. A WDT-caused failover to the backup CPU will be initiated.
Loss of input buffer	No	No	Undetectable Failure	Yes	It is conservatively assumed that a loss of the input buffer causes an undetectable failure (i.e., a toggling signal is still sent to the WDT) and fails the system.
Failure to operate or false operation of solid-state switch	No	Signal Dependent	Signal Dependent	Signal Dependent	A solid-sate switch carries a digital I/O signal of the main CPU. Its failure to operate indicates that the digital signal fails as is. False operation indicates that the digital signal fails to the opposite state. Therefore, based on the normal positions of the solid-state switches defined for each digital signal, the impacts of these failure modes are evaluated using the software and system design information.

Tracking is accomplished in the backup CPU by setting its output to the output from the controllers. Tracking facilitates a smooth transition of control when the backup CPU takes over control after the detection of a failure in the main CPU module.

The description of the main CPU module in Section 3.3.1 applies to the backup CPU module, and they both have the same hardware architecture (Figure 3-2). An FMEA of the backup CPU module was performed similarly to that for the main CPU module, illustrated in Table 3-1. Since the backup CPU module is in tracking mode, all of its failure modes are latent failures except for CCFs. Appendix A presents the detailed FMEA of the backup CPU module.

3.3.3 FMEA of the FWP Controller Module

The FWP controller provides an interface between the main and backup CPUs and the FWP speed controller. It accepts FWP demand signals from both the main and backup CPUs. The FWP controller forwards one of the FWP demand signals (analog signals) from the CPUs based on their status. The operator can use the FWP controller as a manual control station. Other functions of the FWP controller may include monitoring, which is unrelated to the control of the DFWCS.

Figure 3-3 shows the major components of the FWP controller, and indicates that they are similar to those in Figure 3-2 for the main CPU module. The major differences between Figures 3-3 and 3-2 is that the FWP controller is an ASIC-based device and does not have analog and digital backplane buses that might accommodate more I/O devices. The FMEA for the FWP controller module is carried out similarly to that for the main CPU module.

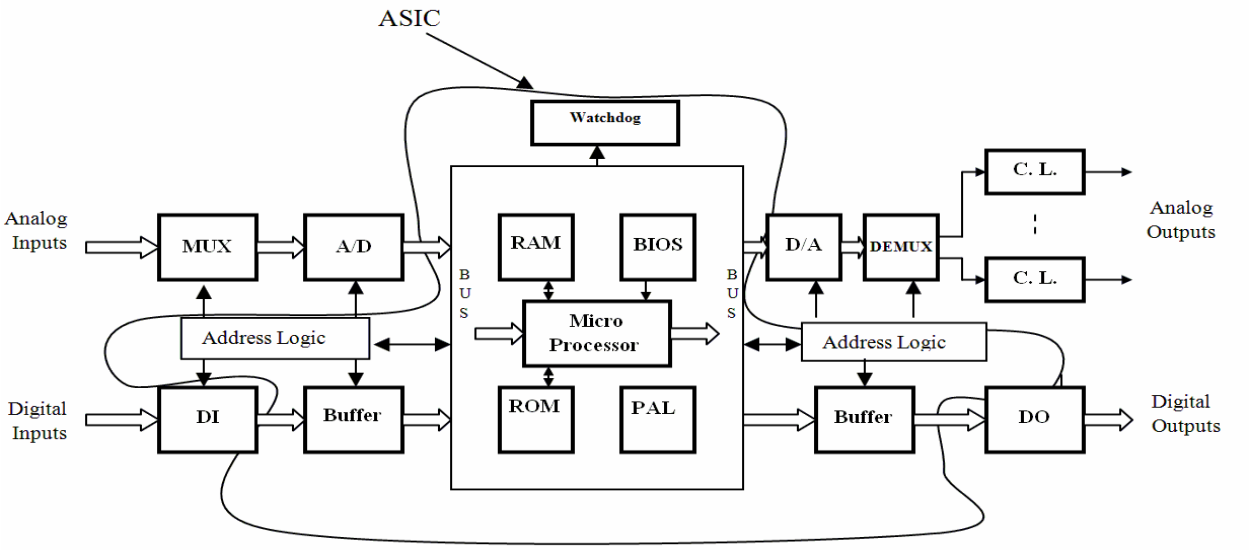

Figure 3-3 Components of the FWP controller module

Since the FWP controller is ASIC-based, some specific ASIC-related failure modes were analyzed. They were the failure modes identified in the hazard analysis of the nuclear power plant. In addition to the component failure modes described in Section 3.3.1, the ASIC-related failure modes of the controller include the following: (1) Loss of power on (PWR_ON) signal that will halt the processor; (2) failure of the display (DISP)-controller, or the DISP-memory is visible in the display (this is only display-related and does not affect controller operation); (3) a fault in the 8051 interface to the display or the 1K dual-ported display memory, causing no writes to

display memory (again, this is only display-related and does not affect controller operation); (4) clock reference failure that will fail the controller; (5) a programmable array logic (PAL) error, which will fail some functions performed by software in the RAM; (6) a loss of RS-485 Jabber (RS-485 Jabber is not used to communicate control-function-related information and does not affect controller operation); and (7) a loss of power supply to the controller, which will cause its operation to fail.

Many failure modes of the internal components lead to a failure of the FWP controller, and, therefore, a failure of the system due to a loss of automatic control of the DFWCS. Some of the FWP controller failure modes related to the backup CPU are latent failures and do not directly affect the operation of the FWP controller. For example, if the demand input from the backup CPU fails, the normal FWP controller function will remain unaffected unless the main CPU has a problem. A few FWP failure modes lead to a failover from the main CPU to the backup CPU and more details can be found in Appendix A.

Table 3-2 summarizes the FWP controller FMEA. Again, the purpose of the table is to illustrate how to perform FMEAs using example failure modes. The detailed version is given in Appendix A.

3.3.4 FMEA of the MFV Controller Module

The hardware of the MFV controller is identical to that of the FWP controller shown in Figure 3-3. An FMEA of the MFV controller is performed similarly to that shown in Table 3-2; hence, it is not discussed here. Appendix A details the FMEA of the MFV controller.

Many failure modes of the internal components lead to failure of the MFV controller, and therefore, to loss of automatic control of the DFWCS. Some MFV controller failure modes related to the backup CPU do not directly affect MFV controller operation, and a few MFV failure modes lead to a failover from the main CPU to the backup CPU. For example, if the demand input from the backup CPU fails, the normal MFV controller function will remain unaffected unless the main CPU has a problem. These failures are latent failures.

In one situation, the FMEA revealed that the system response to a postulated failure mode differs significantly from that indicated by the plant's hazard analysis. When a failed-low MFV demand signal from the main CPU to the MFV controller occurs, the PDI controller should immediately sense the zero output from the MFV controller and take over control by becoming the manual control station for the MFRV. This constitutes a loss of the system (i.e., loss of automatic control), contrary to the description in the plant's hazard analysis which stated that this failure would lead to a failover from the main CPU to the backup CPU, i.e., not a system failure. The hazard analysis does not seem to correctly consider the timing of the associated events. When the failure first occurs, the MFV controller would pass along the failed signal.

The failed signal will be sensed by the PDI controller and sent back to the main CPU by feedback. The PDI controller senses the failed signal immediately, while the feedback to the main CPU leading to a failover has a one-second delay. Therefore, the PDI controller should take over control before there is an opportunity for the failover to the backup CPU.

Table 3-2 Illustrative examples for performing FMEA at component level of the FWP controller module.

Failure Mode	Failure Mode Detected by WDT	Failure Effects on FWP	Fails the DFWCS?	Comments
Software CCF	No	Failed	Yes	It is assumed the CCFs of software or hardware will cause undetectable failure of the FWP controller and fail the entire system.
Hardware CCF	No	Failed	Yes	
The software on the FWP controller seems to be normally running but sends erroneous output	No	Failed	Yes	This is considered an undetectable failure of the FWP controller and will fail the entire system.
Software halt (processor stops updating output)	Yes (flashing display)	Failed	Yes	When the WDT no longer receives toggling signal, a flashing display will result.
The FWP microprocessor seems to be normally running but sends erroneous output (60% of total failure)	No	Failed	Yes	This is considered an undetectable failure of the FWP controller and will fail the entire system.
The microprocessor stops updating output (40% of the total failure)	Yes (flashing display)	Failed	Yes	When the WDT no longer receives toggling signal, it will cause a flashing display.
Loss of PWR_ON Signal	Yes (flashing display)	Failed	Yes	The WDT is out due to the loss of the reset signal from PWR_ON. The processor will halt. The control task stops updating outputs, and the display task stops updating display memory. All the contact outputs will be at the “Open” state. Analog outputs will go to zero mA.
Failure of the display controller or the display memory is visible in the display	Loss of display	Continued Operation	No	This isolated failure has no effect on operation and probably should be excluded from modeling.
A fault in the 8051 interface to the display or the 1K dual-ported display memory which causes no writes to display memory	Loss of display	Continued Operation	No	This isolated failure has no effect on operation and probably should be excluded from modeling.

Table 3-2 Illustrative examples for performing FMEA at component level of the FWP controller module.

Failure Mode	Failure Mode Detected by WDT	Failure Effects on FWP	Fails the DFWCS?	Comments
Clock reference failure	No	Failed	Yes	All functions of the ASIC will stop. The core block (8051 processor) will fail to execute the software. Both the WDT and display will freeze. Analog outputs will drift because the WDT has not expired.
Loss of internal bus	No	Failed	Yes	The I/O of the controller relies on the internal bus. Hence, its loss precludes any processing.
Loss of RAM	No	Failed	Yes	Software must be loaded into RAM to run it. Thus, the application software cannot run upon a loss of RAM.
Loss of BIOS	No	Failed	Yes	The I/O operations of the FWP controller rely on BIOS routines. A loss of BIOS is conservatively assumed to fail the controller and the entire system.
PAL Error	No	Failed	Yes	The failure effects of a loss of the PAL may cause loss of some of the functions performed by the software stored in RAM. This failure is conservatively assumed to fail the RAM and the controller.
Loss of RS-485 Jabber	A DFWCS trouble alarm will be actuated.	Continued normal operation	No	53MC5000 does not use the communication network to transmit control-related information. The failure effects may include loss of warning messages or date and time.
Current loop device fails (drifts) high or fails (drifts) low	No	Signal Dependent	Signal Dependent	1. Both I/O signals can be in the form of current. Failures of different signals have different impacts on the FWP controller and the system. Further analysis is needed for individual signals to determine their impact on the FWP controller module and the DFWCS. 2. Current signals also may drift. It is assumed that they eventually will drift either high or low.

Table 3-2 Illustrative examples for performing FMEA at component level of the FWP controller module.

Failure Mode	Failure Mode Detected by WDT	Failure Effects on FWP	Fails the DFWCS?	Comments
Voltage signal fails (drifts) high or fails (drifts) low	No	Failed	Yes	The only voltage signal is the bias signal from the potentiometer. The FWP controller monitors the rate of the bias change, and should a pre-set limit be exceeded, the FWP controller switches to manual, which is a loss of automatic control and a system failure according to the definition in this study.
Loss of all signals from MUX	No	Failed	Yes	1. The MUX is shared by all analog inputs. Loss of a signal means that the input signal becomes zero. 2. A loss of all signals indicates that the speed-demand signal ANI0 from the main CPU also will fall to zero. The failed signal will be forwarded to the turbine controller. The turbine controller will detect the failure and maintain pump speed at the pre-failure value. This is considered a system failure because of the loss of automatic control.
Loss of one signal from MUX	No	Signal Dependent	Signal Dependent	1. This failure mode indicates a loss of a specific analog signal. 2. Responses to this failure depend on individual signals.
All 16 bits of A/D converter stuck at zeros or ones	No	Failed	Yes	1. Since all analog outputs share the A/D converter, its loss will entail the loss of all AI. If all bits of the A/D converter are stuck at zeros (or ones), all analog inputs are assumed to fail low (or high). 2. The failed speed-demand signal will be sent to the FWP speed controller that will detect the fail-to-low (or fail-to-high) signal, and maintain the FWP speed at its pre-failure value. This is considered a system failure because of the loss of automatic control.
Random bit failure of A/D converter	No	Failed	Yes	Although the processor might detect some random failures, they are conservatively assumed undetectable.

Table 3-2 Illustrative examples for performing FMEA at component level of the FWP controller module.

Failure Mode	Failure Mode Detected by WDT	Failure Effects on FWP	Fails the DFWCS?	Comments
Output of A/D converter fails high or low	No	Failed	Yes	1. Since all analog outputs share the A/D converter, its failure will generate a failure in all outputs. 2. Failure of the A/D indicates a failure of the ANO0 demand signal. The failed signal will be sent to the FWP speed controller that will detect the fail-to-low (or fail-to-high) signal, and maintain the FWP speed at its pre-failure value. This is considered a system failure because of the loss of automatic control.
Drifting output of A/D converter	No	Failed	Yes	It is assumed that the drifted input will eventually drift high or low, and the effects of failure are the same as fail high or fail low, as shown above.
Loss of all output signals from DEMUX	No	Failed	Yes	Loss of a signal means that the signal becomes zero. The DEMUX is shared by all analog output signals.
Loss of one output signal from DEMUX	No	Signal Dependent	Signal Dependent	1. This failure mode indicates a loss of a specific analog signal. 2. Responses to this failure depend on the individual signals.
Loss of address logic	No	Failed	Yes	Loss of address logic is conservatively assumed to be undetectable.
Loss of output buffer	No	Failed	Yes	All digital I/O requires the buffer.
Loss of input buffer	No	Failed	Yes	It is conservatively assumed that a loss of the input buffer will cause a loss of all digital input, and the FWP controller will fail without being detected.
Failure to operate or false operation of solid-state switch	No	Signal Dependent	Signal Dependent	A solid-state switch carries a digital I/O signal. See the discussion of this failure mode for the main CPU module.
Loss of power supply	No	Failed	Yes	1. All analog outputs fail to zero. 2. All digital outputs fail to open status.

3.3.5 Considerations on the BFV Controller and the PDI Controller in the Reliability Model

As discussed in Section 3.2, the BFV controller and the PDI controller are not included in the reliability model of the DFWCS. This section discusses the reasons for their omission.

The BFV controller constitutes the interface between the main and backup CPUs and the bypass feedwater-regulating valve (BFRV). Similar to the MFV controller or the FWP controller, the BFV controller receives the analog demand signals from the CPUs, and passes one of them to the positioner of the BFRV based on the status of the CPUs. The BFV controller can be in automatic or manual mode, and status information from the BFV controller will be sent back to both CPUs. The major signals from the CPUs to the BFV controller are the demand signal and the CPU status signals.

The reliability study of the DFWCS focuses on its operation in the high-power mode. In this mode of operation, the BFRV, controlled by the BFV module, is normally closed. Due to the small capacity of the BFRV, even if the BFV controller fails in such a way that the BFRV is fully open, the DFWCS is expected to easily compensate for this additional feedwater flow. Thus, the failure of analog demand signals from the BFV controller is not expected to significantly affect DFWCS operation.

This study defined the loss of automatic control by the DFWCS as a system failure. Therefore, the failure of Automatic/Manual (A/M) status output from the BFV should be evaluated. According to the control algorithm of the main and backup CPUs, when the BFV A/M status becomes manual[(3)], demand signals received by the CPUs from the MFV and BFV controllers will be sent back to these controllers, respectively. This implies a loss of automatic control of the DFWCS. Hence, the failure of the signal containing the BFV A/M status is relevant to the reliability model. The failure of this BFV A/M status can be accommodated easily in the failure analysis of the BFV A/M status input to the CPUs. Therefore, an explicit BFV controller model is not necessary because the failure of the A/M status can be included in the FMEAs for the CPUs.

The PDI controller normally displays the differential pressure across the MFRV. Its more important function is to monitor the demand output from the MFV controller. If this demand fails to zero, the PDI automatically takes over control from the MFV controller, and becomes a manual control station for the MFRV.

One concern about the PDI controller is whether it can successfully take over the MFV controller when required to do so. However, according to the definition of system failure used in this study, since the PDI controller must be manually controlled after taking over the MFV controller when the MFV demand fails to zero, the takeover of the PDI already denotes a system failure

---

[(3)] For the controlling (Main) CPU, there are two types of tracking in high-power mode: (1) In case of detection of deviation of some signals by the CPU (e.g., the MFV controller demand feedback to the controlling CPU) or indication of CPU failure by the MFV controller, the CPU will enter a tracking mode. In this case, the digital outputs that indicate high-power mode, lower power mode, bypass mode, and override mode all will become false. The CPU will send demand signals received from MFV, BFV, and FWP back to these controllers. (2) If the A/M status of the MFV (or FWP) becomes manual, the CPU will send the MFV (or FWP) the demand signal received from the MFV (or FWP) back to the MFV (or FWP) controller. If the BFV controller becomes manual, the demand signals received from the MFV and the BFV will be sent back to them, respectively. In both tracking modes, the digital status signal of the CPU will not change, e.g., the MFV controller cannot detect that the Main CPU is failed when it is tracking.

due to a loss of automatic control. Therefore, no matter whether the PDI can take over automatically or not, a system failure is considered to have occurred when the MFV demand output fails to zero. Whether the PDI controller takes over or not is no longer relevant in this study.

A second concern is a false takeover of the MFV controller by the PDI when the MFV is normally controlling. This action by the PDI controller will very likely open the MFRV to the maximum, causing an overflow of feedwater and failing the system. A false takeover may be either due to the PDI incorrectly detecting a loss of the MFV demand, or the demand output from the PDI (which will be summed with the MFV demand output) fails to high. Therefore, the impacts of false takeover by the PDI controller are relevant. This failure also can be incorporated into the failure analysis of the MFV controller's demand output since the failure of this input produces the same results.

Given the definition of system failure used by this study and that the DFWCS is considered to be in the automatic high-power mode, the BFV controller and the PDI controller do not necessarily need to be modeled to evaluate the reliability of the DFWCS, provided that the most important failure impacts of their failures, discussed above, are included in the FMEAs of other modules. These FMEAs then can be used to construct the reliability model of the DFWCS. This issue is discussed further in Section 4.4.

3.3.6 FMEAs of Other Components

Sensors and Transmitters

The software of a CPU determines whether sensor inputs are valid by checking for OOR or high rate of change conditions, and uses different logic to process signals accordingly, depending on the type of sensors and the validity status of the signals. This represents the capability of the CPU to detect abnormal conditions of the sensor signals. For example, if the two feedwater flow signals are valid, then they are compared to determine if they deviate significantly. If there is no large deviation, then the average value of the two signals is used in control calculations. If one signal is invalid, then the status of the other CPU is checked. If the status of the backup CPU remains good, the main CPU will fail itself and let the backup CPU take over because the automatic control can still be maintained if the invalidity of the signal is caused by certain failures in the main CPU, not by the sensor or the transmitter.

In some cases, a signal failure may not be detected as being invalid by the CPUs, resulting in a large deviation between the two redundant signals. Since the CPUs share the sensors and transmitters, both CPUs will register a large deviation, no failover will take place, and control will continue with incorrect sensor input. Therefore, the system is likely to have failed. In this study, failure of a sensor is assumed to be detectable by the CPU's OOR detection capability, and the signal is considered invalid. From this assumption, individual sensor or transmitter failures may cause a failover, but will not cause a system failure. This will be further discussed in Chapter 4.

Feedwater Flow Sensor and Transmitters

Loss of one sensor or transmitter, i.e., signal fails high or low (note a loss of the signal would be treated as the signal fails low), will cause the main CPU to failover, and the backup CPU should use the signal from the remaining transmitter after taking control. Loss of both of feedwater flow sensors or transmitters will switch the control from 3-element to 1-element control.

Feedwater Temperature Sensors and Transmitters

Feedwater temperature signals are not used during high-power operation; therefore, their failures do not affect the system. Accordingly, the reliability models do not cover failure of such sensors and transmitters.

Steam Flow Sensors and Transmitters

Loss of one sensor or transmitter will cause the main CPU to failover, and the backup CPU should take over control and use the remaining signal. Loss of both sensors or transmitters will switch the control from 3-element to 1-element control.

Neutron-Flux Sensors and Transmitters

During full power, loss of either sensors or transmitters will be detected, the high-to-low power transfer will be inhibited, and otherwise, the system will continue its operation. Loss of one sensor or transmitter has no effect on system operation.

Steam Generator Level Sensors and Transmitters

Loss of one sensor or transmitter will be detected by both CPUs, and control will continue with the remaining signal. Therefore, loss of both signals is required to fail the system.

DC Power Supplies and 120v AC Buses

The main and backup CPUs each have a dedicated direct current (DC) power supply powered by a 120v alternating current (AC) bus, while the controllers share two redundant DC power supplies each powered by a 120v AC bus. For the CPUs, loss of power supply is indicated by a CPU digital output. Insufficient information is available to determine how this indication is implemented after a loss of power supply although the possible mechanism can be postulated. This digital output has two failure modes, i.e., failure to provide loss of power signal and false generation of loss of power signal. In this study, in order to reduce the total number of failure modes that need to be analyzed, the failure of the CPU power supplies was modeled by adding its failure rate to the latter failure mode, since both of these will have the same effect on system operation. Likewise, the CCF of the DC power supplies and AC buses for the CPUs are included in the CPU CCF event. For the controllers, common-cause failures of the DC power supplies are assumed to be dominant and are explicitly included in the reliability model.

MFV Positioners and Turbine Controller

Both the MFV positioner and the turbine controller are digital devices. For example, the MFV positioner is a microprocessor-based current-to-pneumatic device, and converts the input current signal from the MFV controller to a pressure signal that positions the MFRV. Its failure would result in a system failure. However, these devices were not modeled in this study due to insufficient available information on their design and operation.

3.4 Discussion and Limitations of the Generic Approach

The summarized FMEAs of the DFWCS modules and other DFWCS components afford several observations: (1) many failure modes of components of modules will not fail the system; (2) the impacts of different failure modes for a specific component may be very different from each other; (3) the failure impacts of the same failure modes of the same components on different modules can be significantly different; and (4) fault-tolerance features implemented via specifically designed hardware (e.g., an external WDT) or hardware redundancy (e.g., the main CPU and the backup CPU), or application software, play a vital role in determining the effect of each component failure mode on its respective module and on the entire system. Note, however, that fault-tolerance features may also have a negative impact on the reliability of digital systems if they are not designed properly or fail to operate properly.

The proposed FMEA approach and its implementation make the following simplifying assumptions: (1) drifted analog signals are assumed to eventually drift high or low and can be merged with the failure modes of signal fails high or low, which will be further discussed in Chapter 4, and (2) only one failure mode is assumed for some components, such as the ISA bus, RAM, ROM, BIOS, flash disk, serial port, address logic, and buffer. The only failure mode for these components is the loss of the component. Furthermore, in most cases, their failure impact on the module was considered as an undetected failure due to difficulty in precisely evaluating the impacts. For example, some of the lower level failure modes of memory may be detectable, while some other failure modes are not. This is an issue that can be addressed using the concept of coverage. More detailed modeling, such as through the use of fault injection analysis, as discussed in the next paragraph, is needed to determine if lower level faults can or cannot be detected. While a more systematic treatment of the detectability of component failure modes is desirable, it should also be recognized that detectability of a failure mode is design specific and coverage values obtained for one system will often not be applicable to other systems.

Other assumptions made in this study include: (1) a component can only fail to one of its failure modes and (2) failures of different components are independent of each other whether or not these components are physically wired together, i.e., individual failures are localized. For example, a failure of component A can be propagated to component B to which component A is connected, but this does not introduce a new failure of component B. The former assumption probably can be relaxed by reviewing failure experience and modeling the physics of failure of the components (i.e., considering root causes of failure, such as fatigue and fracture, to study the physical processes that bring about failures), an up-front approach adopted in many countries [Pecht 1994]. The latter assumption is due to lack of design details. If the design details are available, then the assumption may not be necessary because whether or not a failure is localized can be determined manually or by performing supporting analyses using tools, such as fault injection methods. Elks [2008] discussed use of fault-injection method to study the dependability of a digital system by modeling its internal logic in detail, and applied the method to estimate the coverage of the main CPU. This method might be useful for refining the FMEAs of this study. Using the detailed model of a digital system/component considered in a fault injection method, the effects of non-localized failures can be accounted for. The completeness of the failure modes also is an issue. Clearly, the role that failure modes and the associated data play in studies such as this is vital. There are very few public references that describe failure modes of generic digital components and the associated distributions of failure

modes (mainly [RAC 1997b] and [Meeldijk 1996]). Refined definition of failure modes of digital components and associated data are desirable, and further efforts in this area are needed.

Due to the flexibility, variety, and complexity of digital systems, the difficulties in performing FMEAs at the proposed levels also are obvious. The previous description of the FMEA process requires a thorough knowledge of digital systems and their associated components, as well as specific design information on the particular digital system to be analyzed. While it is not a straightforward task to gain a detailed understanding of underlying principles of digital systems/components, i.e., the principles of generic digital components and physical meanings of their failure modes and their potential effects, the more difficult part in the analysis is acquiring and using design information of the specific digital system. The design information is system specific, and must be collected and reviewed extensively to undertake the FMEAs of the system. The FMEAs summarized in Sections 3.3.1 through 3.3.6 were mainly accomplished manually and a significant effort was expended in doing so. The system designers certainly will have the necessary design information but may not perform the detailed analysis performed in this proof-of-concept study. The automated FMEA tool provides an efficient way of making use of the design details to examine system responses to combinations of postulated failures.

The FMEAs for the DFWCS identify the component failure modes that individually result in (or are assumed to result in) system failure (examples are provided in Tables 3-1 and 3-2). However, the sum of the failure probabilities for these failure modes only represents a part of the overall DFWCS failure probability. There are several latent failures for each DFWCS module. Although a single latent failure does not affect system operation, combinations of more than one may fail the entire DFWCS. Therefore, these latent failures necessitate further analyses that may be prohibitive because the impacts on the system of combined latent failures must be assessed and the number of combinations is extremely large. To resolve this problem, Chapter 4 proposes an automated tool that can support FMEAs, given the failure modes of the individual components of the modules. This tool takes advantage of the availability of the source code of the CPUs and controllers.

4. AN AUTOMATED TOOL OF PERFORMING FMEA FOR DIGITAL SYSTEMS

In Chapter 3, an approach was proposed for undertaking failure modes and effects analyses (FMEAs) of digital systems. The approach was illustrated by analyzing the failure modes of individual components of the digital feedwater control system (DFWCS) modules. This chapter describes a software tool that adapts this approach to the DFWCS and automates the process of determining the failure impacts on the system of individual failure modes and of different combinations of component failure modes. The results of the software tool for the individual failure modes were compared with the results of the manually performed FMEA to resolve any differences. The updated FMEA results (for individual failures only) are shown in Appendix A. The FMEA of all the double- and triple-failure sequences was performed using the automated tool, while some random verification was done manually. A failure sequence is defined here as a failure mode of an individual component or a combination of such failure modes that take place in a particular order. The order in which the failure modes occur can make a difference in the effect on the system. The failure sequences that fail the system (which are analogous to "ordered" cutsets) were identified using the automated tool. Although an automated tool is used, the method applied is still referred to as "traditional," since it does not attempt to explicitly model the interactions between the DFWCS and the plant physical processes. The quantification of the failure sequences is discussed in Chapter 7, and is based on failure parameters given in Chapter 6.

4.1 The Advantages of Using an Automated Tool for Evaluating Failure Effects

In the DFWCS, only the central processing units (CPUs) have redundancy, i.e., the main and backup CPUs that use identical software and hardware are redundant to each other. The functions of the controllers primarily are to forward control demands received from the CPUs, and provide some status signals back to the CPUs. Fortunately, the Microlink communication of the DFWCS does not affect the control function of the DFWCS and so does not have to be included in the DFWCS reliability mode; this greatly reduces the effort required to study the couplings and interactions between different modules. Nevertheless, the remaining complexity of the DFWCS raises difficulties in implementing the generic FMEA approach described in Chapter 3. The major difficulties in implementing the proposed FMEA approach are listed below.

1. An in-depth understanding is required of both generic digital systems and the information on the specific software and hardware design of the digital system, as indicated in Chapter 3. Considerable effort is required to gain this knowledge, especially about the specific design of a digital system.

2. Determining the impacts of a specific failure mode on the modules or system is not straightforward. According to the proposed FMEA approach in Chapter 3, the effect of each failure mode on the signal(s) associated with the failed component should be assessed first. Because the components of the entire system are connected by pathways that transfer the signal(s) throughout the system, the responses of the modules and the system to the failure-affected signal(s) must be determined based on detailed analysis of the software and hardware logic, which is a time-consuming process.

3. The system response to a failure also depends on fault-tolerance features that are difficult to capture because they involve the timing of the failure and because signals may be coupled to each other.

4. Even if it were practical to determine the effects of failure of individual component failure modes, there is still the issue of multiple latent failures for each module, as discussed in Chapter 3 and shown in Appendix A. Since a latent failure does not by itself cause system failure, the impact of combinations of failure modes on the system must be evaluated. Considering the number of potential combinations and complexity of interactions between modules, manually implementing the proposed FMEA would be extremely difficult, if not impossible. FMEAs of failure sequences may be more intractable because different orders of failures might entail different system responses.

This study describes an automated tool to support the FMEAs of the DFWCS that offers a practical solution to these issues. In addition, the conceptual development of the automated FMEA tool implementing the approach in Chapter 3 is offered as a general methodology for addressing the complexity of undertaking FMEAs of digital systems in future reliability assessments of such systems.

4.2 An Automated Tool for Evaluating Failure Effects

Essentially, the FMEA tool is a software platform developed from the original source code of the CPUs, and from re-creating the controller software that interfaces with input and output (I/O) variables that represent physical connection signals between the modules, the system, and the controlled process. Inputs to the automated tool are sequences of component failure modes whose effects on the system are determined by the automated tool. To evaluate the impacts of a given sequence, its effects upon associated signals are determined first, based on the FMEA of Chapter 3. For example, component failure modes and their impacts on associated signals are listed in Section 3.3.1 for the main CPU module, and more detailed descriptions of the impacts are given in the comment column of Table 3-1. Then, the software variables representing these signals are modified accordingly and used in simulating the sequence. The simulation propagates the faulted signal(s) of the associated components by executing the software representing the interconnected modules. The impacts of the postulated component failures on the modules and the system are represented by the values of the signals the modules and system process, and therefore, the interactions between the components modeled can be captured by the automated tool. Although the simulation propagates the component failures through all the modules, the module level impacts of Figure 3-1 are not extracted because only system level impacts are of interest. Rules are developed as part of the automated tool, based on the definition of system failure and the status of both CPUs and controllers, such that the system status, i.e., the system response, can be determined automatically. The rules ensure the automatic resolution of whether or not the simulated sequence would result in a system failure.

4.2.1 Scope of the Automated FMEA Tool

The DFWCS consists of the following modules: the main CPU, the backup CPU, the main feedwater valve (MFV) controller, the bypass feedwater valve (BFV) controller, the feedwater pump (FWP) controller, and the pressure differential indicating (PDI) controller. The main and the backup CPUs receive analog input signals from plant sensors and from the controllers. An

external watchdog timer (WDT) monitors each CPU. The digital input signals to each CPU mainly are from the controllers and the other CPUs. The CPUs send analog demand signals and some digital signals to the controllers that, in turn, forward those demand signals to the positioners and the turbine controller, which interpret the demand signals and directly control the main feedwater regulating valve (MFRV), the bypass feedwater regulating valve (BFRV), and the FWP. Chapter 3 explained why it is unnecessary to model the BFV and the PDI controllers. Therefore, the automated FMEA tool consists of the software implementation of the modules for the main CPU, backup CPU, MFV controller, and FWP controller. Also modeled are the functions of the external WDTs for the main and backup CPUs. The scope of the development of the tool also covers the modeling of the failures of sensors and direct current (DC) and alternating current (AC) power supplies.

The FMEA tool is fully automated and able to generate sequences of failure modes, evaluate the responses to them of the components/modules along the signal's pathway, and, ultimately, determine whether system failure occurs. Therefore, developing and applying the automated tool requires: (1) integrating different modules to reproduce all signal pathways, (2) determining the I/O signals of DFWCS modules, (3) establishing a base case using operational data, (4) considering timing issues, (5) defining failure modes using software variables, (6) determining failure effects on modules and the entire system based on system failure criteria consistent with the top event definition, and (7) generating failure sequences. As discussed in Section 4.4, the model does not include the controlled process that interacts with the DFWCS.

4.2.2 Integrating Modules into the Automated FMEA Tool

The automated tool is written in the C language, the same language used for the CPUs, so that the CPU source code can be used directly. The controller software is in a proprietary language that must be converted to C language. Different modules of the DFWCS are integrated into a single software of the automated tool. Although these modules are executed sequentially in the software platform, the order and timing of data exchange are followed as strictly as possible to more realistically simulate the independent execution of software on different processors. Each cycle of the controller software takes 50 milliseconds (ms), and its maximum overrun time does not exceed 110 ms. The cycle time of the CPU software is 100 ms and must not exceed 110 ms. In the automated tool, the controller software are executed every 50 ms, and the control software of the main and the backup CPUs are executed every 100 ms. Figure 4-1 is a flowchart of the automated tool. As assumed in Chapter 3, a failure is permanent and only occurs during the system's steady-state operation. It is noted that the simulation will stop when all outputs (digital and analog) of modules are stabilized after applying the final failure mode.

After starting the simulation, some time elapses before the system initializes all of its modules. Once the system reaches a stable operating point, i.e., a steady state, the simulation of a failure sequence begins. It is noted that the main and the backup CPUs not only obtain input data from the plant and controllers, but they also exchange data. In the real DFWCS system, all of the CPU and controller modules should be running in parallel using the specific data acquired for them. In the integrated automated tool, all of the CPU and controller modules have to be executed sequentially. To mimic the parallel execution of two physical modules and avoid the premature exchange of data, in the simulation the main CPU does not update its outputs until the backup CPU module has been executed, as shown in Figure 4-1. Both the main and the backup CPU application software are run every 100 ms in the automated FMEA tool, and after that, the controller modules are executed.

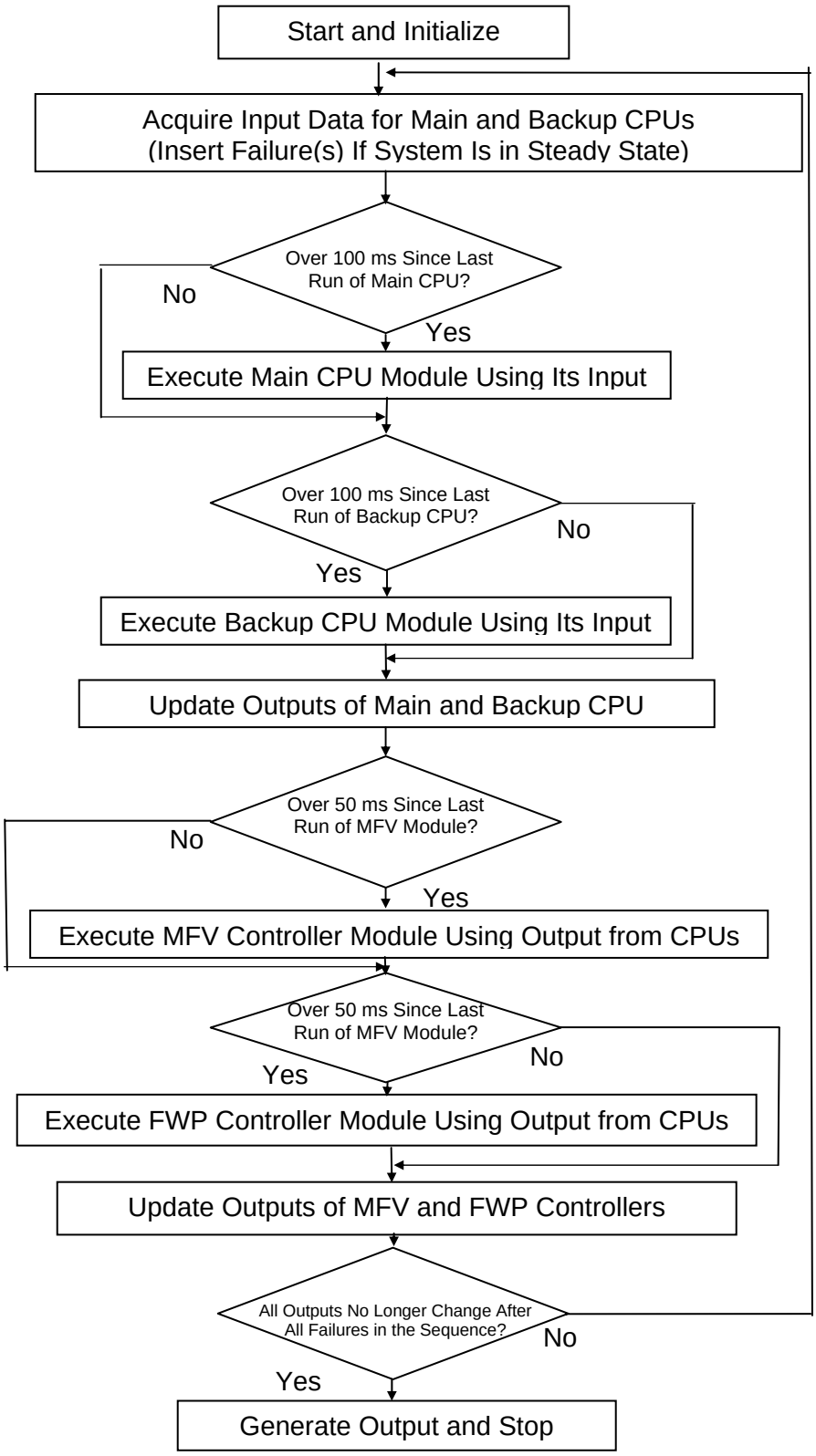

Figure 4-1 Flowchart of the automated FMEA tool

Some updated outputs of the main and the backup CPUs are inputs to the MFV and the FWP controllers, which run sequentially but do not have their outputs updated until after execution of both controller modules, is completed since they also run in parallel physically, as illustrated in Figure 4-1.

After running all of the modules, the system outputs will be examined by comparing them to the corresponding outputs in previous time steps. If the outputs (other than the toggling signals to the WDTs) change, then this indicates that the failure propagation should be continued by repeatedly executing the CPU and controller modules. If the outputs (other than the toggling signals to the WDTs) stay the same for a certain number of consecutive runs of all the modules, a new post-failure steady state is considered to be reached. Whether the system fails can be determined by using a set of defined failure criteria (discussed in Section 4.2.6) to evaluate the final outputs.

4.2.3 I/O Signals of the DFWCS Modules

Another important facet of the automated tool is to determine how component failure modes affect physical signals, and apply the failure modes by modifying software variables representing physical signals. The interconnections between modules are characterized by analog and digital input and digital output signals. The input signals are identified for the DFWCS modules in Tables 4-1 through 4-6, and the corresponding output signals are identified in Tables 4-7 through 4-12.

Analog signals to the CPUs mainly include measurement inputs from the plant sensors and demand signal feedback from the controllers. The analog inputs and analog outputs of the main CPU are identical to those of the backup CPU. The tool uses the same set of sensor input data that represent the plant operating conditions because it does not include a model of the controlled process. The same inputs are used through the entire simulation unless the failure sequence includes a failure of input; then the failure is applied to the specific input.

It is noted that the original software of the CPUs performing the control function does not validate outputs from range checking of output signals. The basis for the output ranges of the main CPU (Table 4-7) is the input ranges for these signals that are fed back from the controller.

If the demand signals to the controllers are out of range, their software simply clamps the values of the signals and continues to forward these demands.

Usually, the digital signals of the original software are used to represent the status information of modules. The only difference between digital signals of the main and the backup CPUs is the CPU identification that basically informs the CPU whether it is the main or the backup. Table 4-1 has detailed descriptions of analog inputs to the CPUs, along with the unit, range, and initial values of the corresponding signals.

The automated tool includes all these signals and their associated pathways. Thus, running the tool ensures that the system response to any failure sequence can be accurately obtained.

Table 4-1 Analog input signals to the main and backup CPUs.

Input	Description	Units	Range	Initial Value	Source of Initial Value
FW Temp1	Feedwater temperature	Degrees F		435.62 °F	8204mn.txt[4]
FW Temp2				435.62 °F	8204mn.txt
FW Pump Bias	Bias for FWP demand	Volts	0V to 5V	−1.05% (converted to 2.47V)	8204mn.txt
OSG Signal	MFV demand from other steam generator (S/G)	Percentage	-25% to 100%	91.40%	8204mn.txt
FWP Signal	Feedback of FWP controller output	Percentage	0% to 100%	Feedback	N/A
LVDT1		Percentage	0% to 100%	87.63%	8204mn.txt
LVDT2				87.63%	8204mn.txt
Feedwater Differential Pressure1	Pressure differential across MFRV	Pounds	0 lb to 300 lb	116.67 lb	8204mn.txt
Feedwater Differential Pressure2				109.05 lb	8204mn.txt
S/G Level1	S/G reservoir level	Percentage (this value is converted from inches)	0% to 100% (at input); during later processing, there is a range of –20” to 20” (53.6% to 75.8%)	0.74” (converted to 65.13%)	Level set-point is selected as level input.
S/G Level2				0.70” (converted to 65.11%)	
Feedwater Flow1	Feedwater flow rate	Amps (A) (this value is converted from a percentage)	0.004A to 0.02A	99.25% (converted to 17.43 milliamperes (mA))	8204mn.txt
Feedwater Flow2				99.15% (converted to 17.40mA)	8204mn.txt

[4] This file contains information on set-points and input which is apparently from a dump of live system data from the plant whose DFWCS was the primary basis for the model developed in this study.

Table 4-1 Analog input signals to the main and backup CPUs.

Input	Description	Units	Range	Initial Value	Source of Initial Value
Steam Flow1	Steam flow rate	Amps (this value is converted from a percentage)	0.004A to 0.02A	99.92% (converted to 17.61mA)	8204mn.txt
Steam Flow2				98.55% (converted to 17.24mA)	8204mn.txt
Reactor Flux1	Reactor neutron flux	Percentage	0% to 125%	99.92%	8204mn.txt
Reactor Flux2				99.70%	8204mn.txt
Level Set-Point	S/G reservoir level set-point	Volts (V) (this value is converted from inches to a percentage, and then from percentage to voltage)	1V to 5V	0.72" (converted to 65.12% and then to 3.60V)	8204mn.txt
BFV Signal	Feedback of BFV controller output	Percentage	0% to 100%	Feedback	N/A
MFV Signal	Feedback of MFV controller output	Percentage	-25% to 100%	Feedback	N/A

Table 4-2 Digital input signals to the main and backup CPUs.

Input	Description	Meaning	Initial Value
BFV Automatic/Manual (A/M)	BFV controller A/M status	true = auto false = manual	True
MFV A/M	MFV controller A/M status	true = auto false = manual	True
FWP A/M	FWP controller A/M status	true = auto false = manual	True
ReactorTrip	Reactor tripped status	true = tripped false = not tripped	False
CPU Identification (CPU_ID)[5]	Main or backup processor	true = main false = backup	True
Turbine Trip	Turbine tripped status	true = tripped false = not tripped	False

[5] The digital inputs to the Backup CPU are identical to those to the main CPU, except that the CPU_ID signal is false.

Table 4-2 Digital input signals to the main and backup CPUs.

Input	Description	Meaning	Initial Value
Main CPU Failed	Main CPU failure status from MFV controller	true = failed false = not failed	False
Backup CPU Failed	Backup CPU failure status from MFV controller	true = failed false = not failed	False
Time Sync	Time synchronization	false = do nothing	False
Bypass Flux1	Bypass Flux1 keyswitch (can be used to manually bypass Flux1 Analog Input	true = bypass input false = do not bypass	False
Bypass Flux2	Bypass Flux2 keyswitch (can be used to manually bypass Flux2 Analog Input	true = bypass input false = do not bypass	False
No Fail in Other	Whether there is failures in the other CPU	true = no failures false = failure	True
Deviation in Other	Whether there is deviation in the other CPU	not used in control software	False
Levels in Other	Whether both S/G level signals in other CPU are valid	true = one or both levels invalid false = both levels valid	False
Flows in Other	Whether steam and feedwater flow rate signals in other CPU are valid	not used in control software	False

Table 4-3 Analog input signals to the FWP controller.

Input	Description	Units	Range, revolutions per minute
Main Pump Demand	Pump demand signal from main CPU	revolutions per minute	0 to 5400
Bias	Bias offset input (from manually controlled potentiometer)	revolutions per minute	-5400 to 5400 (set based on fixed ratio of Bias DC Voltage output in simulation, normally 0)
Backup Pump Demand	Pump demand signal from backup CPU	revolutions per minute	0 to 5400

Table 4-4 Digital input signals to the FWP controller.

Input	Description	Meaning	Initial Value
BackupPwrFail/Test	Backup CPU status indicator	true = No failure false = Failure	True
BackupCpuFail	Backup CPU status indicator	true = Failure false = No failure	False
MainPwrFail/Test	Main CPU status indicator	true = No failure false = Failure	True
MainCpuFail	Main CPU status indicator	true = Failure false = No failure	False

Table 4-5 Analog input signals to the MFV controller.

Input	Description	Units	Range. %
MainDem	MFV demand signal from main CPU	Percentage	-25 to 100
BackupDem	MFV demand signal from backup CPU	Percentage	-25 to 100

Table 4-6 Digital input signals to the MFV controller.

Input	Description	Meaning	Initial Value
BackupPwrFail/Test	Backup CPU status indicator	true = No failure false = Failure	True
BackupCpuFail	Backup CPU status indicator	true = Failure false = No failure	False
MainPwrFail/Test	Main CPU status indicator	true = No failure false = Failure	True
MainCpuFail	Main CPU status indicator	true = Failure false = No failure	False

Table 4-7 Analog output signals of the main and backup CPUs.

Output	Description	Units	Range,%
FWP Demand	FWP demand	Percentage	0 to 100
MFV Demand	MFV demand	Percentage	−25 to 100
BFV Demand	BFV demand	Percentage	0 to 100
TP1	test point 1	N/A	N/A
TP2	test point 2	N/A	N/A

Table 4-8 Digital output signals of the main and backup CPUs.

Output	Description	Meaning
WDT	WDT control signal	toggles to prevent watchdog failure
CpuFail	Whether the system failed	True = not failed False = failed
HiPwrMode	Whether the system is in high-power mode	True = in high power mode False = not in high power mode
Xfering	Whether there is a power transfer	True = transferring False = not transferring
LoPwrMode	Whether the system is in low-power mode	True = in low power mode False = not in low power mode
BfvOr	Whether the bypass feedwater valve is in override mode	True = override mode False = not override mode
DevToPc	Whether there is a deviation to the plant computer	True = deviation False = no deviation
XferInhibit	Whether the transfer is inhibited	True = transfer inhibited False = transfer not inhibited

Table 4-8 Digital output signals of the main and backup CPUs.

Output	Description	Meaning
NoFailures	Whether the CPU has detected a failure	True = no failures False = failure
NoDevs	Whether the CPU has detected a deviation	True = deviation False = no deviations
LvlsGood	Whether both level inputs are valid	True = one or both signals invalid False = both signals are valid
FlowsGood	Whether the steam and feedwater flow rate inputs are valid	True = one or more signals invalid False = all signals are valid

Table 4-9 Analog output signals of the FWP controller.

Output	Description	Units	Range
PumpSig	Pump demand signal	Milliampere	4 – 20 mA
BiasOut	Voltage output level for bias setting	Milliampere	Fixed at 0.5 mA

Table 4-10 Digital output signals of the FWP controller.

Output	Description	Meaning
AMStatMain	A/M status indicator to main CPU	True = Auto False = Manual
AMStatBackup	A/M status indicator to backup CPU	True = Auto False = Manual

Table 4-11 Analog output signals of the MFV controller.

Output	Description	Units	Range,%
MFVSig	MFV demand signal	Percentage	-17 to 100
SgSetpoint (not used)	N/A	N/A	N/A

Table 4-12 Digital output signals of the MFV controller.

Output	Description	Meaning
AMStatMain	A/M status indicator to main CPU	True = Auto False = Manual
AMStatBkup	A/M status indicator to backup CPU	True = Auto False = Manual
BkupCpuFailed	Indicates whether MFV has detected a failure of the backup CPU	True = Failure False = No failure
MainCpuFailed	Indicates whether MFV has detected a failure of the main CPU	True = Failure False = No failure

4.2.4 Establishing a Base Case Using Operational Data

A base case of the DFWCS must be developed that represents the normal operating parameters of the system during full power operation. Although a plant model is unavailable for this study, the base case should normally be created using the operational data from the plant. The "Initial Value" column of Table 4-1 contains a set of data from the operation of the DFWCS in high-power mode (the units of the operating data are given in Table 4-1). The tool must convert them into input signals, whose range is given in the "Range" column of that table, before the CPU software can recognize them. There was no available information about how these conversions are accomplished. By reading the source code of the CPUs, and seeing how the software reads and interprets the input signals, it was determined how plant data are converted to input signals to the software. For example, in the CPU software, the input signals for flows apparently are given in terms of electrical current (between 4 and 20 mA). The software converts flow signals into a percentage using $x_{\%} = \frac{\sqrt{|x_I - 0.004|}}{0.0011676}$, where x_I is the flow signal in Amperes, and $x_{\%}$ is given in the operating data. Therefore, converting the input feedwater and steam flows (in percentages) to quantities read by the software is

$$x_I = \left(0.0011676 \cdot x_{\%}\right)^2 + 0.004$$

For inputs other than flows, where the units are Volts or Amps, the following formulae are used to convert from percentages to the units that the software of the CPUs expects to receive.

FWP Bias: (linear conversion from −100% to 100% ($x_{\%}$) to 0V to 5V (x_V))

$$x_V = 0 + \left(\frac{x_{\%} - (-100)}{100 - (-100)}(5-0)\right) = 5\frac{x_{\%} + 100}{200}$$

S/G Level Setpoint: (linear conversion from 0 to 100% ($x_{\%}$) to 1V to 5V (x_V))

$$x_V = 1 + \left(\frac{x_{\%} - 0}{100 - 0}(5-1)\right) = 4\frac{x_{\%}}{100} + 1$$

In addition, the inputs on the S/G level to the CPUs are in percentages, but the operational data is given in inches, so that the conversion listed below is needed.

S/G Level: (linear conversion from –116.5” to 63.5” (x_{in}) to 0% to 100% ($x_{\%}$))

$$x_{\%} = \left(\frac{x_{in} - (-116.5)}{63.5 - (-116.5)}\right)(100 - 0) = \frac{x_{in} + 116.5}{1.8}$$

These conversions are programmed into the automated tool as an interface between operational data and the CPU software of the DFWCS. The column “Initial Value” of Table 4-1 lists the values of analog inputs to CPUs from the plant. Table 4-2 shows the initial status of the CPU digital signals; the CPU outputs initialize the controllers.

At the start of the simulation, this set of initial values is the input to the main and the backup CPUs. After reaching the corresponding steady state of the system, the failure modes of a sequence are applied automatically by changing the corresponding variables in the software. The simulation will continue to run until the system achieves a steady state.

4.2.5 Timing Issues Addressed in the Automated Tool

This study expended considerable effort addressing problems in timing, including considering execution cycles and built-in delays of the CPU software and controller software, and the order in which failures are introduced. More specifically, the following features also were incorporated in the tool: (1) built-in timers were put in the original source code for the CPUs, such as a 1-second delay for the CPU failover and 10-second delay for CPU initialization; (2) the external WDT of each CPU was modified so it can cause the failover to a healthy CPU if it has not detected the toggling signal from its associated CPU for more than 500 ms; and (3) the flexibility of the tool was extended to permit the application of multiple failures in different orders to evaluate their impacts. These features provide a realistic representation of the DFWCS

performance under failure conditions, given a fixed set of plant sensor input data (since the controlled process is not modeled). Other features of digital systems, such as internal diversity and self-healing, in general, can also be captured when developing the FMEA tool.

4.2.6 Criteria for Automatically Determining System Failure

As discussed in Chapter 1, failure of the DFWCS is defined as a loss of automatic control; Chapter 3 defined the failure modes of DFWCS modules. Accordingly, system failure can be defined in terms of the states of these modules. In the DFWCS, the automatic control (demand calculation) is performed by the main or backup CPU. During normal operation, one of the CPUs is controlling and the other is tracking. The system becomes "failed" if a controller switches to manual mode or the demand output from a controller is incorrect. In some cases, based on the manually performed FMEA, an individual failure directly results in a system failure and does not need to be evaluated/simulated using the FMEA tool. Based on the definition of system failure and an understanding of DFWCS operation, a set of rules was created for the tool to automatically determine whether a system failure occurs given each sequence of failures.

The DFWCS is considered failed if any of the following conditions is encountered:

1. Both the main and the backup CPUs are failed. When both CPUs fail, the system will fail due to loss of automatic control. The DFWCS may or may not automatically detect failures of the CPUs; in either case, the system must be manually operated.

2. The main and the backup CPUs have been tracking for longer than one second. When both CPUs are in tracking mode, the controllers switch to manual mode, resulting in a loss of automatic control.

3. The MFV controller and/or the FWP controller switch to the manual mode, resulting in a loss of automatic control.

4. The MFV controller and/or the FWP controller stop using the demand signal (based on the CPU status signals they receive) from the controlling CPU for at least one second[(6)]. Thereafter, the controller uses demand signals from the tracking CPU, instead of the controlling CPU. A CPU in tracking status simply forwards the demand signals received from the controllers back to them again, which means automatic control is lost.

5. The demand output of the FWP controller fails either low or high. This is a conservative rule based on assuming that the turbine controller will take over the FWP controller immediately after it detects the fail high or fail low of the FWP demand. This is particularly relevant for the latter event because it will cause the pump to fail, and needs to be rectified immediately.

6. The MFV controller output fails low, causing a takeover by the PDI controller. The PDI controller becomes a manual control station after this automatic takeover, thereby resulting in a loss of automatic control.

[(6)] The reason to have this rule is illustrated using the FWP controller as an example. If one of the main CPU status signal inputs to the FWP controller changes to "failed," the FWP controller will stop using the demand signal from the main CPU and take the demand signal from the backup CPU. However, the main CPU is still in controlling mode because it does not know that the FWP considers it failed. The backup CPU is in tracking mode and simply passes the FWP demand signal received from the FWP controller to this controller again. This is a loss of automatic control and thus a system failure.

4.2.7 Generation of Failure Sequences

As indicated in Chapter 2, the system is considered initially to be in a state where all components of modules are normal, i.e., not failed. Each component failure mode of a failure sequence makes the system transit to a new system state. The FMEA tool can automatically determine whether a system state fails the DFWCS or not using the rules described above.

The order of failures is important because different orders may entail different impacts on the system, as discussed in Section 4.3; hence, the order should be followed strictly in generating failure sequences. An individual failure that does not fail the system constitutes the first failure in a double-failure sequence. The second failure can be any individual failure modes of a different component. Similarly, triple sequences arise from adding one of the individual failure modes of a different component as the third failure of a double combination that does not fail the system. It is not necessary to consider additional component failure modes for double sequences that fail the system. This same process can be followed for obtaining sequences containing higher number of failures. Verification of the completeness of failure sequences generated in this way is straightforward.

One concern with the above process is that the state space of the DFWCS is huge considering the possible number of combinations of the failure modes of all components of the modules. It is impractical and unnecessary to generate and/or evaluate all possible failure sequences because expectedly the probability of a failure sequence will decline as the number of failures contained in the sequence increases. Introducing a convergence criterion will determine whether the failure sequence generation process can be stopped after evaluating the sequences that contain a certain number of failures i.e., whether the probability of those sequences that do not fail the system is significantly smaller than the system failure probability calculated from the sequences containing fewer failures.

The numbers of generated single, double, and triple sequences are 421, 128,779, and 36,844,679, respectively. Clearly, evaluation of the double and triple sequences is impossible without the automated FMEA tool.

As stated previously in Section 3.3, it is assumed in this study that a component can only fail once in a given failure sequence, i.e., after one failure modes of the component has occurred, other modes cannot occur for the same component. This assumption is believed to hold for most of the digital components because available information on digital component failures seems to suggest so, and is believed to be more realistic than assuming that components can always fail more than once. It may be possible that a certain component fails to an intermediate failure modes before it reaches one of the other failure modes. If recognized, such a sequence of failures would need to be considered when generating the failure sequences and can still be simulated using the FMEA tool. Furthermore, the resulting failure sequence or sequences can also be quantified using the Markov method discussed in Chapter 5.

4.2.8 Validation of the Automated FMEA Tool

Before the automated tool was developed, an FMEA for individual failure modes was performed manually. The results of the automated tool for individual failure modes were compared with the results of the manual FMEA to resolve any differences. In some cases, based on the manual FMEA, an individual failure directly results in a system failure, e.g., a loss of Industry Standard Architecture (ISA) bus, and does not need to be evaluated/simulated using the FMEA tool.

There is no need to validate this type of failure sequence. The correctness of the system failure criteria defined in Section 4.2.6 was checked by comparing the manual FMEA to that performed using the automated tool for all individual failures and for some double- and triple-failure sequences. The updated FMEA results (for individual failures only) are shown in Chapter 3 and Appendix A. If the tool is to be used for a regulatory application, it would need to be subjected to systematic verification and validation.

4.2.9 A Summary of the Automated Tool Development and Illustrative FMEA Examples

A summary of the procedure to develop and implement the automated tool is provided in Figure 4-2. Note, Step 3 has to be performed manually and is critical in the tool development, because after it is finished and coded in the automated tool, the faulted signals will be propagated by the tool automatically based on the results of Step 3 for given sequences. Also note, for detailed models of complex systems, the number of failure sequences generated using the automated tool may become unmanageable. In these cases, establishing a convergence criterion could be useful to limit the number of sequences generated. This is discussed in more detail in Chapter 7.

For the purpose of better understanding of the automated FMEA tool, two examples provided in this section are used to explain how the component failure(s) propagate through the connections and how the system failure can be determined automatically by the automated FMEA tool.

The first example is a single failure that leads to system failure. The failure mode "MfvDI-CCI2-NCFO-" represents a normally-closed-fail-open failure of a DI CCI2 to the MFV controller from the main CPU. The failure causes the MFV digital input "MainCpuFail" (see Table 4-6) to change to 0, which indicates the failure of the main CPU. As a result, the MFV digital output "MainCpuFailed" (see Table 4-12), which is sent to both the main and the backup CPUs, changes to 1. Thereafter, the main CPU enters the tracking mode and the backup CPU takes over the control. These changes cannot take effect immediately due to the fixed execution cycle of software. The MFV controller still temporarily passes demand from the tracking (main) CPU, but quickly takes demand from the backup CPU, and automatic control of the MFV controller resumes. However, the main CPU digital output "CpuFail" (see Table 4-8), indicating the status of the main CPU, does not change to notify controllers of its failure, as designed. The FWP controller still passes the demand signal from the tracking (main) CPU. Rule No. 4 in Section 4.2.6 becomes applicable and the automated FMEA tool records this component failure mode as a system failure.

Another example is a double sequence that causes system failure. The first failure is a latent failure "FwpDI-CCI2-NCFC-," i.e., a normally-closed-fail-closed failure of the digital input CCI2 to the FWP controller. CCI2 ("MainPwrFail/Test," see Table 4-4) informs the FWP controller of the main CPU status. The system continues its operation with this latent failure until the occurrence of the second failure "Mn-MuxFwFI1LOS—." The second failure is a loss of the main CPU multiplexer signal representing the feedwater flow input #1 "FW Flow1" (see Table 4-1). This failure causes the following changes in the main CPU digital outputs: (1) "NoDevs" (see Table 4 - 8) changes to 1, indicating that a deviation between the two feedwater flow input signals has been detected; (2) "NoFailures" (see Table 4-8) changes to 0, indicating a failure in the main CPU; and (3) "FlowsGood" (see Table 4-8) changes to 1, indicating that not all flow input signals are valid. Accordingly, these signals go to the backup CPU and the changes are reflected in the changes of the corresponding digital inputs (see Table 4-2) to the backup CPU,

i.e., (1) “Dev in Other” (changes to 0); (2) “No Fail in Other” (changes to 1); and (3) “Flows in Other” (changes to 1). The main CPU digital output “CPUFail” (see Table 4-8) changes to 0, indicating that the main CPU fails and starts tracking. The MFV controller passes demand temporarily from the tracking (main) CPU, but it will also pass the failure status of the main CPU to both main and backup CPUs via the digital output signal “MainCpuFailed.” The backup CPU will take over the control upon receiving the signal. The MFV controller starts passing demand from the controlling CPU, i.e., the backup CPU. However, the FWP controller is still passing demand from the main CPU that is tracking because the first failure, “FwpDI-CCI2-NCFC-” indicates that the main CPU is still controlling. Rule No. 4 is again applicable and the automated FMEA tool records this failure modes sequence as a system failure.

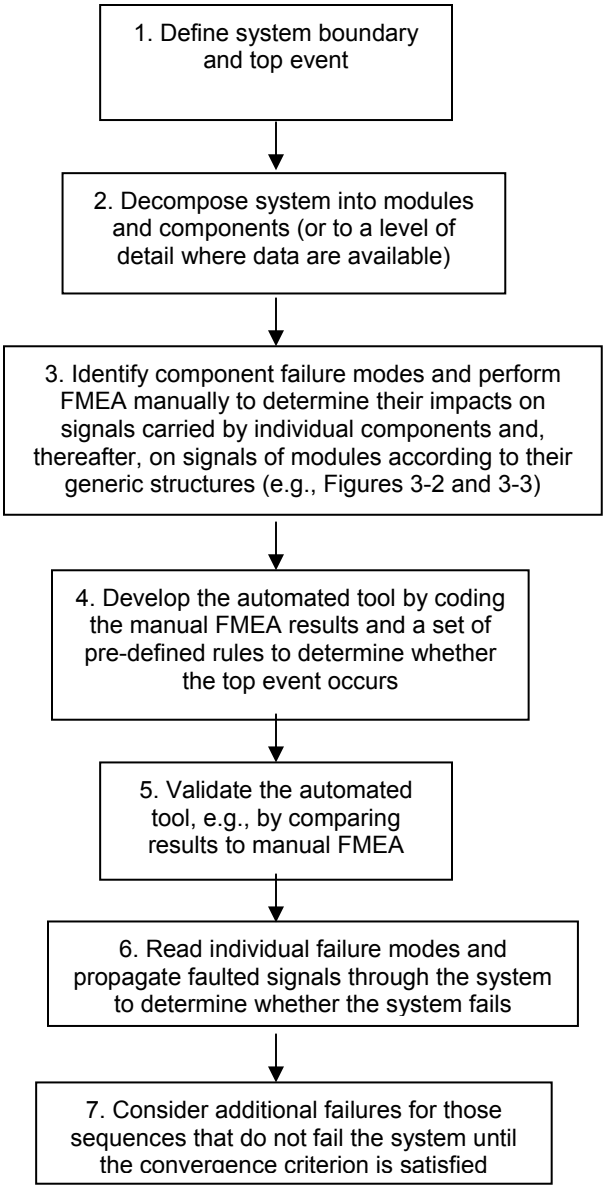

Figure 4-2 A summary of the automated FMEA tool development and implementation.

In addition, it is evident that many common-cause failures (CCFs) are single failures, such as CCFs of the CPUs and controllers, and power supplies for the CPUs and controllers. In this study, CCF of power supplies of the CPUs are included as part of the CCF of the CPUs, “CCFCCFCPU--Fail-.” “CCFCTRPwr--Fail-” represents the CCF of controller power supplies only. Due to a lack of redundancy of the MFV and FWP controllers, many failures related to them are also single failures. These include losses of clock reference signals, ISA buses, buffers, RAMs of the MFV and FWP controllers (represented by “MfvClk-----Loss-,” “FwpClk-----Loss-,” “MfvISABus--Loss-,” “FwpISABus--Loss-,” “MfvBufOut--Loss-,” “MfvBufIn---Loss-,” “FwpBufOut--Loss-,” and “FwpBufIn---Loss-,” respectively). Some failures of the main CPUs are also single failures, such as losses of buffer and flash disk (represented by “Mn-BufIn---Loss-” and “Mn-FlsDisk-Loss-,” respectively).

4.3 Findings Using the Automated FMEA Tool

The automated FMEA tool considers the timing and order of failures. The importance of the latter was recognized using the tool. In simulations, some failure sequences did not cause system failure, but the same set (or a sub-set) of component failures in a different order did result in system failure. For example, the FMEA of the main CPU indicates that the main CPU digital input containing the MFV A/M status (which is normally closed) failing open is a single failure. The failure causes the main CPU to receive a signal that the MFV is in manual status which causes the main CPU to enter the tracking mode, and represents a loss of automatic control, i.e., a system failure. On the other hand, if a failure that causes a failover of the main CPU to the backup CPU occurs first, then the single failure of the main CPU digital input of the MFV A/M status does not affect the system because the main CPU no longer is the controlling CPU. Hence, considering the number of individual failure modes that cause the main CPU to change from controlling to tracking mode, there should be many double (or triple) sequences that contain one of these single failures as the second (or the third) failure and that will not fail the system.

As another example, consider a double sequence consisting of two failures, fail out-of-range high (OORH) of one feedwater flow analog input to the main CPU (“Mn-AI-Fwfl1OORH”), and all-bit stuck at 1 of the Analog/Digital (A/D) converter of the backup CPU (“Bk-AD-All—OORH”). Neither one of the two failures would cause the system to fail. If “Bk-AD-All—OORH” occurs after “Mn-AI-Fwfl1OORH,” the system fails because an OORH failure of the feedwater flow input to the main CPU will entail a failover to the backup CPU, and this, in turn, will be failed by its A/D converter failure, eventually failing the system. Reversing the order of this double sequence, the backup CPU will be failed first and the response of the main CPU to the failure “Mn AI Fwfl1OORH” is to use the other feedwater flow input; it will not attempt to failover to the backup CPU because the main CPU knows its failure status. There are 510 double sequences of this type.

A potential weakness of the DFWCS was identified using the automated tool. That is, an incorrect main CPU status from the MFV controller to the CPUs causes a loss of automatic control. It was anticipated that such a failure would only cause a failover from the main CPU to the backup CPU. The failure is assumed to be a localized failure at the output circuit of the MFV controller, which causes an incorrect failed status of the main CPU be sent to the CPUs, while the MFV controller is still aware of the correct status. The main CPU will enter tracking mode upon receipt of the signal without failing itself. The backup CPU will think that it is in control and send its calculated demand signals to the controllers. Since the MFV controller still

considers that the main CPU is in control, it continues sending the signal from the main CPU which is in tracking mode. Effectively, the automatic control is lost.

4.4 Discussion and Limitations of the Automated FMEA Tool

There are obvious advantages in using the automated FMEA tool to support an FMEA. An automated process of generating sequences of failures, applying them to the system, and determining the system status affords a systematic, reliable, and fast way of supporting an FMEA. The tool automatically addresses interactions between modules or components that are difficult to thoroughly evaluate manually. The tool also can consider issues related to timing and ordering of failures, as discussed in Section 4.2.7. In addition, although full power operation is assumed in this study, the automated FMEA tool should also be applicable to low power mode, if some changes in failure criteria are made.

The automatic tool has limitations. The first is that it is difficult to preserve all of the timing features of the system. As indicated in Section 4.2.2, the execution cycles of the software are variable. The software of controllers are started every 50 ms. However, in reality, this 50 ms cycle is not fixed and should be adjusted by the actual time it takes to run the software, which is unknown. The maximum overrun time is limited to less than 110 ms. This variable execution cycle of the controller software is difficult to reproduce in the automated tool; it probably can be considered a trivial issue based on assuming that the controller does not need to adjust the cycle unless something very unusual occurs.

Another timing issue, which is perhaps more important, is associated with the time when an additional failure occurs given one or more failures have taken place and the system has not failed yet. It is assumed that the system is in a steady state before any failure occurs. If the additional failure occurs after the control system and the controlled processes have again reached steady-state condition after the transient caused by preceding failure or failures, then the automated FMEA tool can correctly determine the system response. If the additional failure occurs before the system reaches a steady state subsequent to the preceding failure(s), the impact of the additional failure on the system cannot be captured by the FMEA tool, because the FMEA tool does not have a model of the controlled process and is not able to determine the transient response. However, it is expected that the duration of the transient subsequent to the postulated failure or failures is very short comparing with the duration of one year, and the occurrence of the additional failure during the transient is very unlikely, given the assumption that the failures are independent of each other. Ignoring the transient period should not have a significant impact on the results.

The second limitation concerns the usage of the developed automated tool to perform the system FMEA without including the dynamics of the controlled process. The DFWCS interacts with the controlled feedwater process via analog signals only, i.e., measurements from the plant are sent to the DFWCS and the demand signals are sent to the regulating valves and pumps. Therefore, digital signals are not directly related to the controlled process, and digital interactions (mainly between different modules of the system) are well captured in the automated tool. For analog signals, it is almost certain that the failure modes of fail high or fail low also can be captured due to the range and validity check of the analog signals in the software. For example, if the demand output signal of MFV fails high or low, the MFRV's responses eventually will cause overflow or underflow of feedwater and fail the system, no

matter how long it takes. The impacts of the fail high or fail low of MFV demand output can certainly be captured. The only concern here is the failure modes of signal drifting.

Since this study considers that the components of the DFWCS are not repaired during power operation, a drifting signal will not be corrected, and so its long-term impacts are treated simply, i.e., it is assumed that a drifted signal eventually will move OORH or out-of-range low (OORL). However, this assumption may be conservative for some cases where the extent of the signal drift is small, since plant information indicates that the control system may be able to cope with a small amount of drift. Nevertheless, in other cases, this assumption may be nonconservative. For example, the failure mode of OORH or OORL of the analog input of the S/G level signal to the main CPU will be detected and the remaining good signal will be used if the failure is that of a S/G level sensor or transmitter, allowing the normal operation of the system to continue. However, if this signal neither drifts to OORH or OORL, i.e., it drifts, but within the range, the system might fail due to the undetectable bad signal. Therefore, the assumption that the signal will ultimately drift OORH or OORL is nonconservative in this case.

This is a limitation of the modeling approach used in this study that can potentially be addressed in the future by refining the definition of drifting failure modes into two types, within and outside the range, and (1) including and accounting for the failure modes of drifting within the range in the automated tool or (2) including plant dynamics (i.e., incorporating a model of plant response) to simulate the impacts of such failure modes. Including plant dynamics could help capture the subtle timing aspects of the performance of the DFWCS. However, this issue is likely to be difficult to address even with a model of the plant included in the automated tool, because the failure impacts are affected not only by how the signal drifts, but also by the system operating point when the failure occurs. A subtle deviation in the drifting signal may cause completely different responses. In addition, it is not clear, at present, whether the increased accuracy of modeling obtained through incorporation of a plant dynamics model would justify the increased complexity and effort required for intensive simulation.

A third limitation is related to the translation of the controller software, that was written in a proprietary language, to the C language used in the simulation tool. The translation may be subject to errors or loss of details. Due to the simplicity of the controller software, this is not expected to be a significant issue for this study.

The automated FMEA tool can be enhanced by defining more detailed failure modes for certain components, such as RAM. The major challenge with RAM is not how to incorporate the failure modes in the tool but the limited understanding of their failure modes. For example, some of the lower level failure modes of memory may be detectable, while some other failure modes are not. This is an issue that can be addressed using the concept of coverage, as discussed in Section 3.4. Other potentially achievable improvements of the tool development are associated with the components or modules which are modeled in a less detailed manner or not modeled at all in this study. These components and modules include the external WDTs of the main and backup CPUs, the BFV controller module, and the PDI controller module, which are discussed below.

The WDT monitors the toggling signal from a digital output of the main or backup CPU and sends out the status signal (digital) of the main or backup CPU to the MFV, BFV, and FWP controllers. In both the FMEAs and the automated FMEA tool development, the functions of the WDTs are considered (e.g., identification of the WDT-detectable failures), while the failure modes of the WDTs, which could be either a failure to indicate the failure status of the

associated CPU when the CPU has failed or a spurious signal output indicating that the CPU has failed when it has not, are not modeled due to a lack of design information of the WDTs, as stated in Chapter 2. The failure modes of the WDT correspond to the failure modes of failure to operate and false operation of a solid-state switch. Therefore, the effects of the WDT failure modes can be accounted for in the FMEA of the digital input of the main and backup CPU status signal sent from the WDT to the controllers. The reliability model quantification needs data for the WDT failure modes, which can only be obtained from a detailed analysis to identify the WDT component failures that may cause the WDT failure modes.

The failure modes of the BFV and the PDI controllers that are relevant to but not included in the reliability model of the DFWCS were identified in Chapter 3. The failure modes are associated with the BFV A/M status output to the CPUs and the PDI controller demand output, which is added to the MFV controller demand as the input to the positioners. Similar to the consideration of the WDT failures, the impacts of the relevant failure modes can be simply accounted for in the CPU modules or the MFV controller FMEAs of the signals that are related to these failure modes, i.e., the digital input of the BFV A/M status to the CPUs and the demand output of the MFV controller. For the purpose of this proof-of-concept study, the quantification of these two failure modes was limited to a single component failure each. To be complete, however, a detailed analysis of the failures of the components contained in the BFV and the PDI controllers would be needed to determine if there are other component failures or combinations of component failures (in the BFV or PDI modules) that could also result in one of these two failure modes.

4.5 General Discussion on Developing Automated FMEA Tools for Digital Systems

The development of the automated FMEA tool represents a general methodology for addressing the complexity of undertaking FMEAs of digital systems in future reliability assessments of such systems. However, while the process for developing and using the automated FMEA tool is generic, the tool itself is application specific, since it is based on the source code of the system being modeled.

Use of the automated FMEA tool reduces the role of the analysts to just performing the component-level FMEAs manually and to verifying the results of the automated tool. The automated tool also facilitates identification of potential design weaknesses, as indicated in Section 4.3. Another important advantage of using the source code to build the FMEA tool is to preserve the fidelity of the original software, making the resulting reliability model a more realistic representation of the system. The concept of FMEA tool development can be applied to study the reliability of highly integrated digital systems.

Some potential difficulties in applying the approach for developing automated FMEA tools are discussed below.

- It is desirable to use the source code which should be available to the nuclear power plant but may not be available to the United States Nuclear Regulatory Commission or its contractors. If the source code is not available, an FMEA tool can still be developed using design information, such as a functional description of the software, although the tool will not be as realistic and may not be suitable to be used to study a system in detail.

- In some cases, the source code may be written in a language that has to be translated into the one used in developing the automated tool. Care has to be taken in such translation because of potential misinterpretation of the original software logic. The proposed approach for developing an automated FMEA tool does not call for modeling of the controlled processes. This limitation does not appear to be too strict, as discussed in Section 4.4. In particular, for protection systems, it may not be necessary to model the controlled processes, because once a protection function is actuated, the protection system has accomplished its function, i.e., feedback from the plant may not need to be considered. However, for digital control systems, it is still uncertain as to whether it is necessary to include a model of the controlled processes.

- The proposed FMEA approach may require that a very large number of sequences be evaluated using the automated tool. The computational effort required may be tremendous, especially if one has to integrate multiple, interactive digital systems in the analysis. However, it should be recognized that the proposed approach inherently is capable of parallel processing because determining failure effects of different sequences are not related to each other and can be processed independently. Therefore, a linear scalability of simulation can be achieved by distributing the sequences onto multiple computers, and the results can be collected and combined. This offers a practical solution for the complexity and scale of digital systems. Another option is to simplify the model by grouping failure modes together, as proposed in NUREG/CR-6962 [Chu 2008a].

5. MARKOV MODEL OF DIGITAL FEEDWATER CONTROL SYSTEM

A Markov model of a system models the transitions among system states in terms of transition rates that typically represent the occurrences of failures and repairs. The Markov model can be illustrated by a transition diagram consisting of system states and transitions among them that represent failures and repair rates. It also can be expressed by a set of differential equations associated with the transition diagram, as taken from Equation (1-2) in Chapter 1.

$$\frac{d\underline{P}}{dt} = \underline{M}\,\underline{P}$$

where $\underline{P}$ represents the probabilities of the system states, and $\underline{M}$ is the transition matrix containing the constant transition rates among the system states. The solution of the differential equations, with the initial condition that the system is in a successful state, probabilistically denotes the temporal behavior of the system. For example, the sum of the probabilities of success states is the system reliability, from which the frequency of system failure is calculated, as taken from Equation (1-1) in Chapter 1.

$$f = -\ln[R(T)]/T$$

where f is the frequency of system failure, T is the time period, and $R(T)$ is the reliability within T or one minus the probability of system failure by T. As discussed in Section 1.3, the frequency is the average frequency over the period T.

This chapter documents the development of a Markov model of the digital feedwater control system (DFWCS). The top event is the loss of automatic control of the feedwater system. The development builds upon the failure modes and effects analysis (FMEA) and simulation tool discussed in earlier chapters. In particular, the FMEA identifies the failure modes of the components of the system, and the simulation tool identifies those individual failure modes and combinations (sequences) of failure modes that entail system failure. Chapter 6 discusses the failure parameters used to quantify the Markov model.

The following considerations significantly affect the development of the Markov model.

- All components, including those playing a standby role, e.g., the backup central processing unit (CPU), are operating at all times and can fail at any time.

- Typically, a component can have more than one failure modes with different effects that must be modeled differently. A component is assumed to fail only once in a given failure sequence, i.e., after one failure mode of the component has occurred; other modes of the same component cannot take place. More discussion is provided in Section 3.3.

- In evaluating the effects of sequences of failure modes, the order in which failures take place is recognized to affect the impacts on the system. Therefore, order must be accounted for in developing the model, i.e., in defining possible transitions out of a system state and their end states. More discussion is provided in Section 4.2.7.

- Since the model was developed to assess the frequency of an initiating event, the plant is assumed to be in the mode of power operation. In this mode of operation, it is expected that if some components of the system fail, they will not be repaired because this activity would likely cause or require a reactor trip. Hence, the plant staff would wait until the reactor has been tripped for another reason to carry out any needed repair. For this reason, it was considered that components of the system cannot be repaired or replaced while the system is operating.

The sections of this chapter document the process of developing the Markov model. Appendix C contains more description of Markov modeling, along with the detailed analytical solution of the Markov model for the DFWCS.

The Markov method can be used to identify the significant contributors to a digital system's failure probability in two main steps: (1) quantifying the sequences that fail the system by applying this method, and (2) calculating the contribution of each failure mode in the sequences to the probability of failure of the system, similar to the calculation of standard probabilistic risk assessment (PRA) importance measures, such as Fussell-Vesely.

5.1 Development of a Markov Transition Diagram

Chapter 3 identifies the failure modes of the components of the system, including those of the support systems, i.e., the direct current (DC) power supplies and 120v alternating current (AC) buses. The failure modes of each component define its possible states. System states are defined as the combinations of component states, including the order in which the components fail. Order is explicitly modeled in defining the possible transitions out of a system state.

A transition diagram of the system is developed, starting from a system state in which every component is in perfect condition. The possible transitions out of this state are all of the failure modes of the components of the system. Each such failure modes would lead to a different system state that may or may not be a failed state. If a state is a failed state of the system, then it becomes an absorbing state, i.e., a state with no transition out of it. If a state does not correspond to system failure, then additional failure modes of components are considered as possible transitions out of the system state that engender additional system states. The above process continues and the transition diagram grows to form a tree, until all the end states of the tree are absorbing states. Graphically, the development of the transition diagram is described below.

It is assumed that there are M components and each component, i, has $N_i, i \in [1, M]$ failure modes (states) that are represented as $C_{(i,j)}, i \in [1, M], j \in [0, N_i]$. It also is assumed that $C_{(i,0)}, i \in [1, M]$ represents the component's normal state, i.e., there is no failure of component i. As discussed in Section 3.4, the components are assumed to be independent of each other, i.e., their failures are independent, as illustrated in Figure 5-1.

Thus, the system states can be represented by combinations of the states of individual components. The Markov model we are interested in is one wherein the system starts from a state with no component failure, i.e., the initial system state is $C_{(1,0)} C_{(2,0)} \cdots C_{(M,0)}$; the transitions to other states that contain component failures are characterized by the Markov model shown in

Figure 5-2. Let $\lambda_{(i,j)}$ be the failure rate of failure modes j of component i. Each additional failure generates a new system state; the order of failures should be strictly followed when generating failure sequences in this model because differences in the order may produce different results.

Figure 5-2 shows that there is no component failure in Layer 1, one in Layer 2, and M failures in Layer (M+1). Generally, a fully expanded Markov model would consist of all possible combinations of component failures in all possible orders, as indicated in this figure. The transition diagram expands very quickly with increasing number of components and component failure modes. In practice, a system state that represents system failure can be made an absorbing state without further expansion. This consideration drastically reduces the size of the transition diagram, such that the model becomes manageable.

Understanding the notations of system states in Figure 5-2 is very important, wherein components with failures always appear before those without failures, and the failures that appear first are the ones that occur earlier, e.g., there are two failures in the system state $C_{(i,j)}C_{(1,N_1)}C_{(2,0)}\cdots C_{(i-1,0)}C_{(i+1,0)}\cdots C_{(M-1,0)}$ with the *j-th* failure modes of component i occurring first, followed by the failure mode N_1 of component 1; no other components are failed in this system state.

An illustration of a small portion of the Markov transition diagram for the DFWCS is shown in Figure 5-3. A description of the event identifiers in the figure can be found in Appendix B.

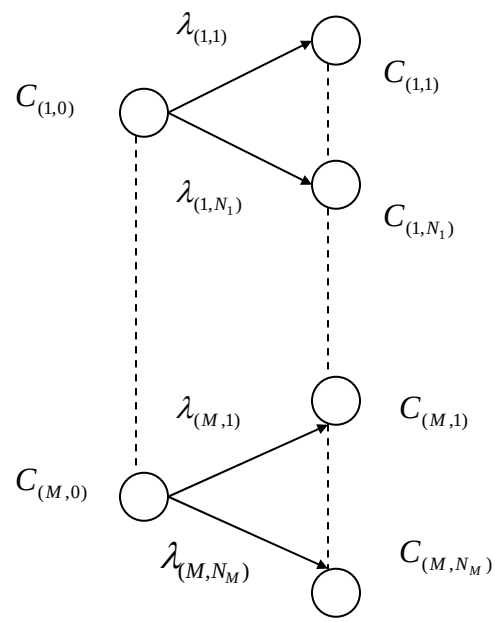

Figure 5-1 Markov models for *M* independent components

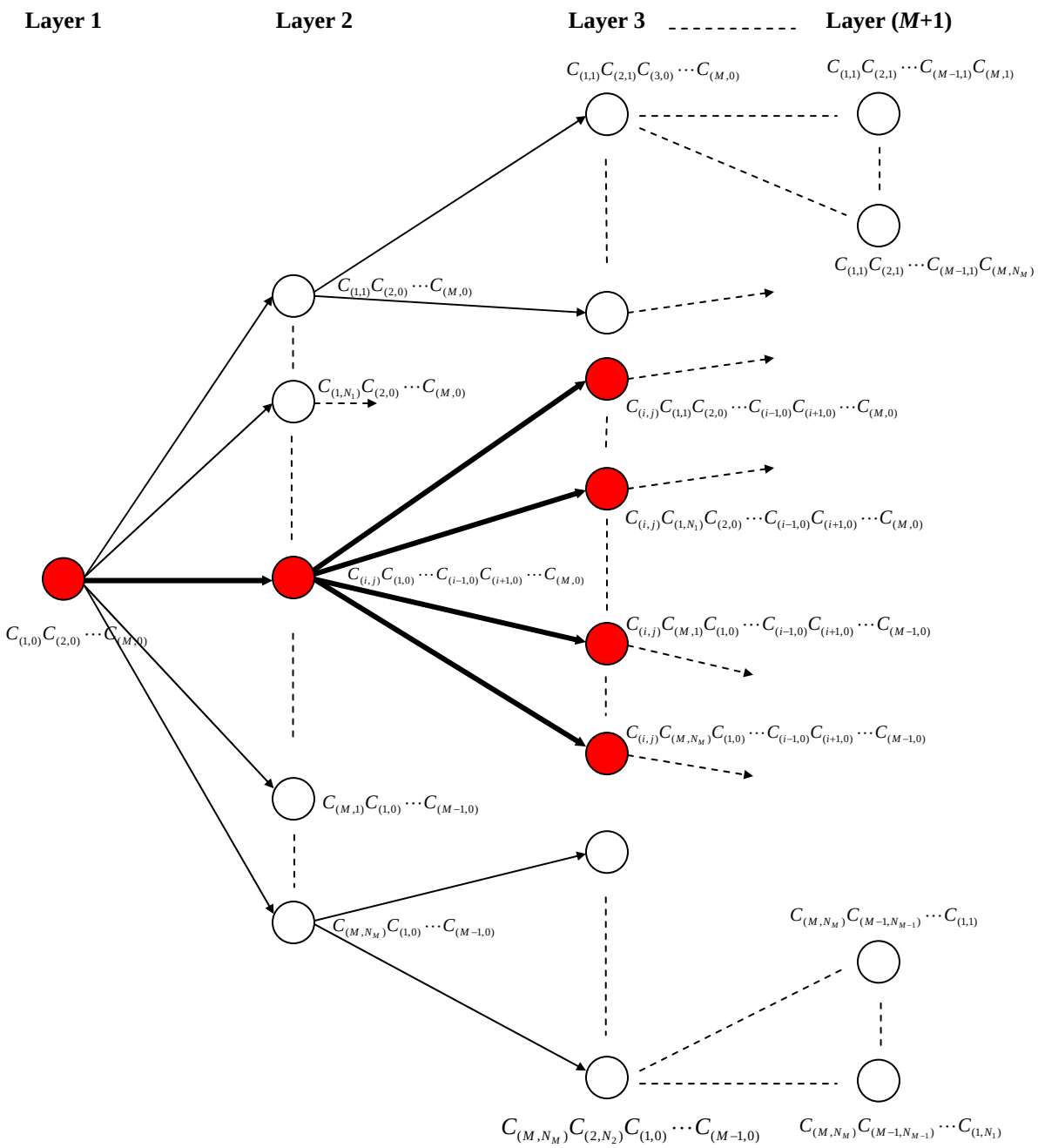

Figure 5-2 Markov model of a system with *M* components

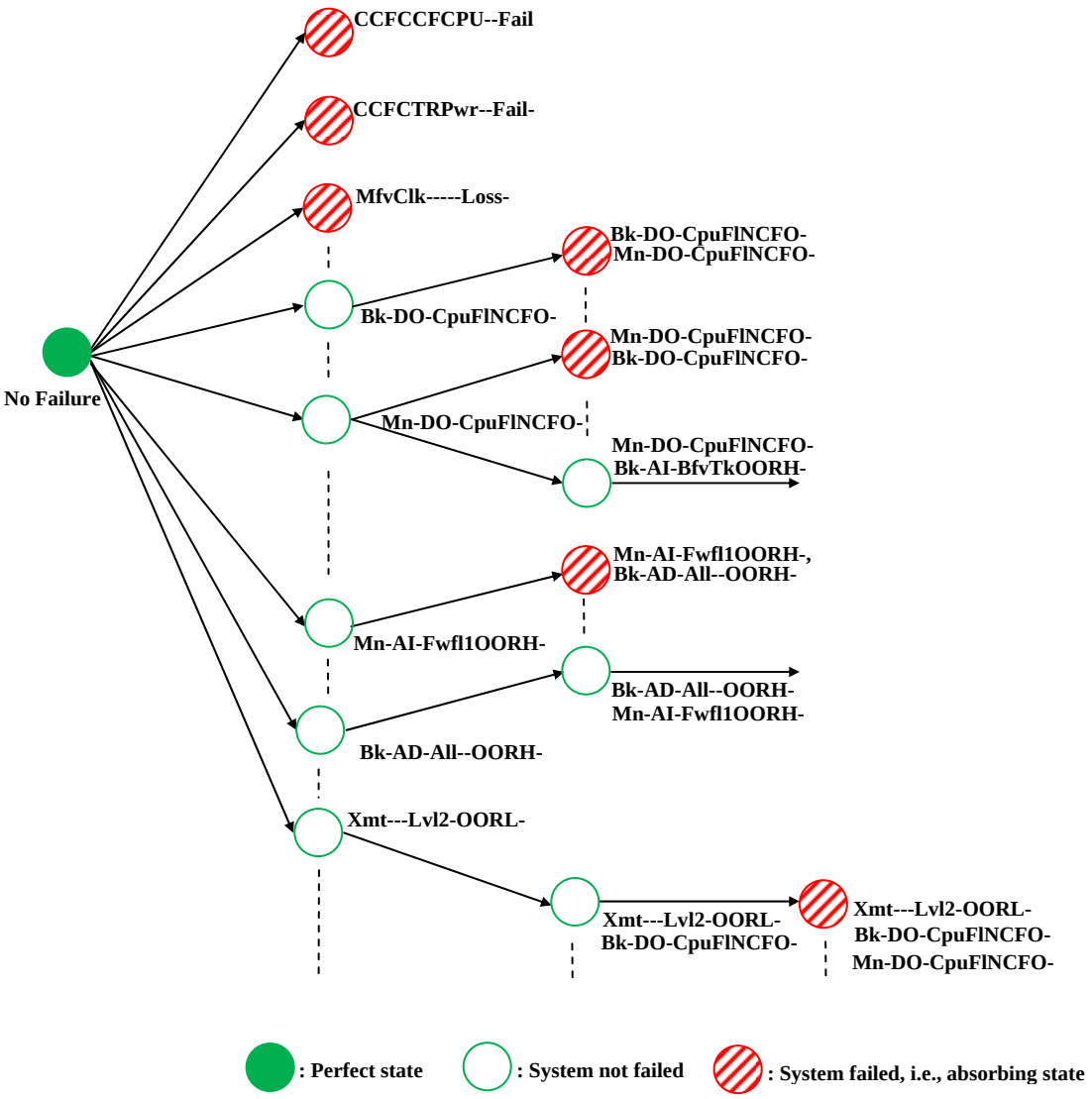

Figure 5-3 A small portion of the Markov diagram for the DFWCS

For the Markov model of the DFWCS system, the total number of the individual failures (i.e., the states in Layer 2) is 421, of which 112 (e.g., common-cause failure (CCF) of the CPU or controller modules or failure of the clock reference of the MFV controller) directly lead to a system failure and thus become absorbing states in Layer 2. Each of the rest of the individual failures (i.e., those in Layer 2 that do not directly lead to system failure) are then individually paired up with every other individual failure to expand the Markov diagram to Layer 3. The "paired" failures in Layer 3 that result in system failure become absorbing states. Following the same procedure, the Markov diagram can be continuously expanded to Layers 4, 5, and so on, as needed, by adding individual failure modes to sequences that do not fail the system. Note that sequences with the same failure modes, but in a different order, lead to different Markov states since they might have different impacts on the system. As discussed in Section 4.3, for

two individual failures, namely, failure out-of-range high (OORH) of one feedwater flow analog input to the main CPU ("Mn-AI-Fwfl1OORH") and all-bit stuck at 1 of the analog/digital (A/D) converter of the backup CPU ("Bk-AD-All—OORH"), neither of them would cause the system to fail by itself. If "Bk-AD-All—OORH" occurs after "Mn--AI-Fwfl1OORH", then the system fails, and in the opposite order, the system continues its operation with these latent failures.

In this study, the CCF of the CPU (or controller) modules is treated as a failure of a "pseudo-component" that contains all of the major components of a CPU (or controller) module. The failure rate of the CCF was calculated by adding the failure rates of the failure modes of all components contained in the "pseudo-component" and multiplying the sum by a beta factor. It is assumed that the CCF causes system failure, which is conservative because not all of the failure modes included in the pseudo-component cause system failure. Other CCF events, e.g., some sensor CCFs, that do not fail the entire system, are further expanded in the Markov diagram. More discussions on CCF modeling can be found in Section 6.3.

5.2 Analytical Solution of the Markov Model

The structure of the transition diagram in Figure 5-2 is in the form of a tree. Therefore, the associated differential equation can be solved sequentially from left to right. That is, the equation for the node with every component in good condition can be solved first, and the solution substituted into the equations for the states immediately to its right; thereby allowing the equations to be solved. The process continues along each branch of the tree until an absorbing state is reached.

Let P and $\dot{P}$ represent the probability and its rate of change, respectively, of a state of the Markov model of Figure 5-2. The following differential equations can be written for the first two states of the shaded branch of the transition diagram by inspecting Figure 5-2:

$$\dot{P}_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} = -\sum_{u=1}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)}P_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} \tag{5-1}$$

$$\dot{P}_{C_{(i,j)}C_{(1,0)}\cdots C_{(i-1,0)}C_{(i+1,0)}\cdots C_{(M,0)}} = \lambda_{(i,j)}P_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} - \sum_{\substack{u=1\\u\neq i}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)}P_{C_{(i,j)}C_{(1,0)}\cdots C_{(i-1,0)}C_{(i+1,0)}\cdots C_{(M,0)}} \tag{5-2}$$

where the second term in the right side of Equation (5-2) represents transitions from the state $C_{(i,j)}C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}$ to all of its associated states in Layer 3.

In general, for a given system state consisting of a sequence of *k* component failures, i.e., $C_{(i_1,j_1)}C_{(i_2,j_2)}..C_{(i_M,j_M)}$, $i_k \in [1,M]$, $j_k \in [0,N_{i_k}]$ with $k = 1,2,..M$, and $j_k \neq 0$ and $j_{k+1} = \cdots = j_M = 0$, we have

$$\dot{P}_R = \lambda_{(i_k,j_k)}P_S - \sum_{\substack{u=1\\u\neq i_1\\\vdots\\u\neq i_k}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)}P_R\,, \tag{5-3}$$

where P_R is the state $C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}$ with k failures, and P_S is the state preceding P_R.

Letting

$$p_0 = \sum_{u=1}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)} \text{ and } p_l = \sum_{\substack{u=1\\ u\neq i_1\\ u\neq i_2\\ \vdots\\ u\neq i_l}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)} \text{ for } l = 1,2,\cdots,k \text{, Equations (5-1) to (5-3) become}$$

$$\dot{P}_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} = -p_0 P_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} \ , \tag{5-4}$$

$$\dot{P}_{C_{(i_1,j_1)}C_{(1,0)}\cdots C_{(i-1,0)}C_{(i+1,0)}\cdots C_{(M,0)}} = \lambda_{(i_1,j_1)}P_{C_{(1,0)}C_{(2,0)}\cdots C_{(M,0)}} - p_1 P_{C_{(i_1,j_1)}C_{(1,0)}\cdots C_{(i-1,0)}C_{(i+1,0)}\cdots C_{(M,0)}} \ , \tag{5-5}$$

and

$$\dot{P}_R = \lambda_{(i_k,j_k)}P_S - p_k P_R \ . \tag{5-6}$$

As mentioned previously, the equations can be solved sequentially, i.e., solving Equation (5-1), substituting the solution into Equation (5-2), and then solving Equation (5-2), and so on.

This process continues along each branch of the transition diagram until an absorbing state is reached. For an absorbing state with k failures, the second term on the right-hand side of Equation (5-3) becomes zero. It easily can be demonstrated that the solutions of absorbing states with one, two, and three failures are

$$P_{C_{(i_1,j_1)}C_{(2,0)}\cdots C_{(M,0)}} = \frac{\lambda_{(i_1,j_1)}}{p_0}[1-e^{-p_0 t}] \ , \tag{5-7}$$

$$P_{C_{(i_1,j_1)}C_{(i_2,j_2)}C_{(1,0)}\cdots C_{(M,0)}} = \frac{\lambda_{(i_1,j_1)}\lambda_{(i_2,j_2)}}{p_0 p_1}[1-\frac{p_1}{p_1-p_0}e^{-p_0 t} - \frac{p_0}{p_0-p_1}e^{-p_1 t}] \tag{5-8}$$

and

$$\begin{aligned} P_{C_{(i_1,j_1)}C_{(i_2,j_2)}C_{(i_3,j_3)}C_{(1,0)}\cdots C_{(M,0)}} = & \frac{\lambda_{(i_1,j_1)}\lambda_{(i_2,j_2)}\lambda_{(i_3,j_3)}}{p_0 p_1 p_2}[1-\frac{p_1 p_2}{(p_1-p_0)(p_2-p_0)}e^{-p_0 t} \\ & -\frac{p_0 p_2}{(p_0-p_1)(p_2-p_1)}e^{-p_1 t} - \frac{p_0 p_1}{(p_0-p_2)(p_1-p_2)}e^{-p_2 t}] \end{aligned} \tag{5-9}$$

Equivalently, the solution of the differential equations can be obtained using Laplace transforms. It is proven by induction in Appendix C that, in general, for a system state consisting of a sequence of k component failures, the solution of Equation (5-3) in the Laplace transformed space is

$$P_{C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}}(s)\Bigg|_{\substack{j_k\neq 0\\ j_{k+1}=0\\ \vdots\\ j_M=0}} = \frac{\lambda_{(i_1,j_1)}\lambda_{(i_2,j_2)}\cdots\lambda_{(i_k,j_k)}}{(s+\sum_{u=1}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})(s+\sum_{\substack{u=1\\ u\neq i_1}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})\cdots(s+\sum_{\substack{u=1\\ u\neq i_1\\ u\neq i_2\\ \vdots\\ u\neq i_k}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})} \quad (5\text{-}10)$$

Undoubtedly, if $j_M \neq 0$, i.e., all components of the system are failed in a certain way, Equation (5-7) becomes

$$P_{C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}}(s)\Big|_{j_M\neq 0} = \frac{\lambda_{(i_1,j_1)}\lambda_{(i_2,j_2)}\cdots\lambda_{(i_M,j_M)}}{(s+\sum_{u=1}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})(s+\sum_{\substack{u=1\\ u\neq i_1}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})\cdots(s+\sum_{v=1}^{N_{i_M}}\lambda_{(i_M,v)})s} \quad (5\text{-}11)$$

Furthermore, if the expansion of the Markov model is stopped such that the number of failures contained in end states is k, the probability of system state $C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_k,j_k)}C_{(i_{k+1},0)}\cdots C_{(i_M,0)}$ for $j_k \neq 0$ and $j_{k+1} = 0$, which then becomes an end state, is given by

$$P_{C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_k,j_k)}C_{(i_{k+1},j_{k+1})}\cdots C_{(i_M,j_M)}}\Bigg|_{\substack{j_k\neq 0\\ j_{k+1}=0\\ \vdots\\ j_M=0}} = \frac{\lambda_{(i_1,j_1)}\lambda_{(i_2,j_2)}\cdots\lambda_{(i_k,j_k)}}{(s+\sum_{u=1}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})(s+\sum_{\substack{u=1\\ u\neq i_1}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})\cdots(s+\sum_{\substack{u=1\\ u\neq i_1\\ u\neq i_2\\ \vdots\\ u\neq i_{k-1}}}^{M}\sum_{v=1}^{N_u}\lambda_{(u,v)})s} \quad (5\text{-}12)$$

It should be noted that the poles of Equations (5-7) and (5-9) always are distinct under the assumption that a component only fails once. Therefore, the corresponding time domain solution of the equations easily can be expressed in terms of poles of Equation (5-7). The probability of state $C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}$ with $j_k \neq 0$ and $j_{k+1} = \cdots = j_M = 0$ is given by

$$P_{C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}}(t)\Bigg|_{\substack{j_k\neq 0\\ j_{k+1}=0\\ \vdots\\ j_M=0}} = \sum_{l=0}^{k}[(s+p_l)P_{C_{(i_1,j_1)}C_{(i_2,j_2)}\cdots C_{(i_M,j_M)}}(s)\Big|_{s=-p_l} e^{-p_l t}] \quad (5\text{-}13)$$

It can be shown readily that Equation (5-10) leads to Equations (5-4) to (5-6) if k is set to 1, 2, and 3, respectively. The use of the solution in quantifying the top event, i.e., loss of automatic control of the DFWCS, is described in Chapter 7.

Note, if the components can fail multiple times, some of the poles of Equations (5-7) to (5-9) might be the same. In this situation, the time domain solution cannot be calculated using Equation (5-13) and the numerical inverse Laplace transform has to be used instead as presented in Section C.2.4 of Appendix C.

5.3 A Simplified Markov Model

Assuming the component failures are rare, i.e., the probabilities of failures are small, the full Markov model described in the previous sections can be simplified by ignoring competition among the failure modes, i.e., for a sequence that causes system failure the probability that no other failures took place is assumed to be 1. This can be called rare event approximation, i.e., the failures are rare and non-occurrence of other failures in a failure sequence definition can be assumed. Figure 5-4, below, represents the Markov model of such a sequence.

If the failure rates are numerically different, then the simplified Markov model can be easily solved from the solution of the full Markov model, i.e., by setting the poles in Equations (5-4) to (5-6) and (5-10) to the individual failure rates, e.g., $p_0 = \lambda_{(i_1, j_1)}$. If the failure rates are identical, as expected in some cases, then a numerical method can be used to solve the Markov model; Appendix C provides more details.

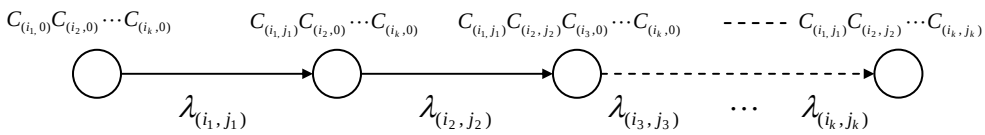

Figure 5-4 Markov model of a system with *k* components and each component has one failure mode

This approximate method should produce a reasonable result if the top event is a rare event, as is expected to be the case for a reactor protection system. For the DFWCS, whose failure is not very rare, the approximate method may not produce good enough results. Note that this model is the same as the typical fault tree method for quantifying initiating event frequencies when modeling a system consisting of components in parallel, but accounting for the order of component failure occurrences.

5.4 Discussion and Limitations of the Markov Model

Due to the level of detail considered for the DFWCS (i.e., many low level components are considered, each with a few failure modes and possible component states), it is not practical to consider all possible system-level states that can be defined in terms of component-level states, i.e., the possible system-level states are too numerous. This state explosion issue is addressed by deriving an analytical solution of the Markov model and then considering dominant contributors/sequences of the system, using a concept similar to that of cutset truncation that is typically done in a PRA. That is, those system states with a larger number of component failures tend to have a lower probability of occurrence than those system states with fewer component failures. In developing the Markov transition diagram of the system, system states are defined starting with the state in which every component is in perfect condition. Additional system states are defined by assuming individual component failure modes take place, each bringing the system to a new state with one additional failure. Successively, system states with one, two, three, and a higher number of failures can be defined. In general, the process would generate all possible system states and is subject to the state explosion issue. Quantification of the system states is done to calculate the system failure probability, during the expansion of the

transition diagram, and the expansion of the transition diagram is terminated when convergence in the calculated system failure probability is achieved (as previously discussed in Section 4.2.7, and further discussed in Section 7.1).

In this study, it is assumed that a component can only fail once and the analytical time domain solution of a Markov state can be obtained under this assumption. For a CCF that does not fail the entire system, e.g., some sensor CCFs, the correct approach to avoid the violation of the assumption is to expand Markov state of the CCF by adding failures of only components that are not contained in the "pseudo-component" representing the CCF. However, this was not done in the study and should be accounted for in future studies.

An important assumption of the Markov model described in this chapter is that repair is not possible, which is the case for the DFWCS. For other digital systems, such as a reactor protection system, on-line repair may be possible, and the analytical solutions of this chapter cannot be used. If repair of components can be done with the system operating, the Markov model has to be modified by adding transitions that represent repairs, making it much more difficult to solve. Using the simplified Markov model described in Section 5.3, the governing equations in the Laplace transformed space can be solved analytically, and the inverse Laplace transform can be solved in the same way described in Section 5.3. The accuracy of the simplified Markov method needs to be further explored and if necessary better approximate methods can be developed. Alternatively, since repair for digital systems may likely occur at a level higher than the components included in this study (e.g., at the circuit board level), it may be possible to model the system at this higher level. At the higher level, due to its reduced complexity, it may be practical to solve the model (including repair) numerically.

6. ESTIMATION OF FAILURE PARAMETERS

In this chapter, estimates are given of the parameters needed in developing and quantifying the reliability model of the digital feedwater control system (DFWCS). They include the failure rates of the components of the system, the percentage breakdowns of each individual failure rate into its constituent failure modes (failure-mode distributions), and the parameters of common-cause failure (CCF). The data used in this study are based on the analysis of NUREG/CR-6962 [Chu 2008a] which reviewed publicly available databases and performed a Bayesian analysis that attempts to account for variability of different raw data sources. In the review, potential weaknesses and limitations of the available databases were identified and discussed, and no attempt was made to validate or invalidate the available databases. The limitations in the publicly available failure parameters of digital components identified in NUREG/CR-6962 [Chu 2008a] indicate that additional research and development is needed in this area. The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).

Chapters 3 and 4 describe the process used to identify the failure sequences that lead to DFWCS failure, using a failure mode and effects analysis (FMEA) and a simulation tool. To arrive at a system failure frequency, the failure modes and sequences must be quantified in terms of their failure rates using the data available. An important reason underlying the generic FMEA approach proposed in Chapter 3 is the availability of failure data for generic digital components. Reliability prediction methods (RPMs), such as those of the Military Handbook 217F [Department of Defense 1995], Telcordia SR-332 [2001], and PRISM [Reliability Analysis Center (RAC) PRISM] are the only public reliability databases that provide failure parameters and raw data for the components of digital systems. They have weaknesses; the estimates may not be accurate enough due to use of conservative assumptions and lack of applicable data [Gu 2007, Pecht 1994] or applicable to the system being analyzed (e.g., the DFWCS in this study), and do not address uncertainties. NUREG/CR-6962 [Chu 2008a] provides more discussion on RPMs. Section 6.1 of this report describes how different sources of information, including the raw data of PRISM [RAC PRISM], were used in a Hierarchical Bayesian Method (HBM) analysis [Atwood 2003] estimating the failure rates needed for the DFWCS model. Application of the HBM is complicated by lack of information about the raw data and obtained population variability curves with very large uncertainties. The curves are used in this benchmark study to exercise the models and should not be used in quantifying models developed to support decision-making.

The failure rate of a component usually includes contributions from all of its failure modes. The detailed FMEA analyses in Appendix A reveal that different component failure modes may entail very distinct failure effects on the system. Therefore, component failure rates must be split into the failure rates of the individual component failure modes, i.e., expressed in terms of the distributions of failure modes that break down failure rates into the contributions of the failure modes. Section 6.2 describes how the failure modes distributions were estimated from available information.

Section 6.3 discusses modeling of CCFs, including those of the central processing units (CPUs), sensors, transmitters, direct current (DC) power supplies, and alternating current (AC) buses.

Software failures are beyond the scope of this study. In developing the reliability model of the DFWCS, placeholders for including software failures were identified, and they were assigned very low failure rates (arbitrarily chosen as 10^{-08} per hour) so that they would not impact the quantification results of the benchmark study.

Table 6-1 lists all the failure parameters used in this study including the uncertainty parameters. It is re-emphasized that the data are used in this benchmark study to exercise the models and should not be used in quantifying models developed to support decision-making.

6.1 Hierarchical Bayesian Analysis of PRISM Data and Failure Rates from Other Sources

To properly estimate the component/system-specific failure parameters, it is desirable to have such data for the specific digital components and systems of interest. In reality, the specific failure data often is unavailable, as is the case in this study, necessitating the use of failure data or parameters of similar components. The HBM analysis offers a way of using generic data of similar components collected from different sources to estimate a distribution representing the variability among the different sources, i.e., a population variability curve. This curve can represent a generic distribution for the parameter of interest, and be further used as the prior distribution in a simple Bayesian updating using component-specific data. The method can be considered as a generalization of the common two-stage Bayesian analysis [Kaplan 1984] by imposing higher order in its hierarchical structure, i.e., having more than two stages. Its application in this study is the same as that of the two-stage Bayesian analysis.

Yue [2006] earlier employed the HBM, and documented it in NUREG/CR-6962 [Chu 2008a], to assess the generic failure rates of a spectrum of digital components using raw data taken from the PRISM database [RAC PRISM]. These raw data are expressed as the number of failures in a number of hours. They were collected from different sources, i.e., from different manufacturers, designs, quality levels, and environments. The sources of the PRISM database [RAC Manual] are not clearly specified, and only identified in terms such as "…warranty repair data from a manufacturer." PRISM [RAC PRISM] further categorizes the failure records of a specific type of component, e.g., memory, according to (1) sub-level component types, e.g., random access memory (RAM) or programmable read only memory (ROM); (2) quality, e.g., commercial grade or military grade; (3) environment, e.g., ground or airborne; (4) hermeticity, e.g., plastic or ceramic; and (5) time within which the data are collected. Before applying the HBM, Yue [2006] grouped the failure records of different qualities, environments, hermeticities, and periods.

In the HBM application, failure rates were assumed to be lognormally distributed while the hyper-priors were assumed to be uniformly distributed. The upper and lower bounds of the hyper-priors were selected such that they covered the resulting posterior distributions of the hyper-parameters. NUREG/CR-6962 [Chu 2008a] discusses issues related to using the HBM application to estimate failure parameters; they will not be further described here.

Due to the large variability in the sources of the data, the resulting population variability curves have large uncertainties.

Some components of the DFWCS modules do not have failure rate estimates from the HBM analysis performed for this study because either they are not digital components, or no PRISM data was found. The failure rates of these components are obtained from either the RACRates

model of PRISM, i.e., an RPM method of PRISM [RAC PRISM], or from other sources of reliability data. The components in the former group include analog/digital (A/D) and digital/analog (D/A) converters, current loops, and solid-state switches. In these cases, there was no uncertainty information and so an error factor of 5 was arbitrarily assumed.

Alternatively, failure rates of some components, including sensors of flux, flow, and level and their transmitters were taken from Savannah River Site (SRS) [Blanton 1993].

Failure data for some components are available in other sources as well as in the PRISM database [RAC PRISM]. For example, the failure rate of a Multiplexer (MUX) and Demultiplexer (DEMUX) is given in both PRISM and Aeroflex [2005]. The failure rates in the latter were adopted here because Aeroflex presents them as different failure modes, unlike PRISM which gives only an aggregated failure rate for MUXs/DEMUXs.

Table 6-1 details the failure rates of different components of the DFWCS, together with their corresponding failure modes distributions and CCF factors which are described in Sections 6.2 and 6.3, respectively. The error factor is defined as the square root of the ratio of the 95th and 5th percentiles of the assumed lognormal distribution. As can be seen in Table 6-1, the distributions are very broad for some of the component failure modes. The effects of varying loads and operating environments may be factors that contribute to the large uncertainty in these failure rates.

It should be pointed out that the state of knowledge in understanding the failure modes and estimating the failure probabilities associated with digital instrumentation and control (I&C) is currently very limited. The ever-changing technology in manufacturing the digital I&C components could eliminate some failure modes, add other failure modes, and significantly improve the equipment reliability by making them more resilient against some stressors. Therefore, it is reasonable to assign larger uncertainties to the failure rates for digital I&C components.

6.2 Failure Mode Distributions

A failure modes distribution represents how a component failure rate should be broken down into the failure rates of individual failure modes. For example, in this study, the failure rate of a level sensor has a failure modes distribution of 20% out-of-range high (OORH) and 80% out-of-range low (OORL), based on the failure modes/mechanism distributions of the RAC [1997]. The sources of the failure modes distributions that could currently be found and were used in this study are summarized below. They may not exactly match the components of the DFWCS modules but are the best approximation presently found; and their completeness and accuracy awaits validation. For the purpose of addressing the uncertainty associated with the failure mode distributions, it was assumed they are uniformly distributed within the arbitrarily assumed upper and lower bounds specified in Table 6-1.

Software failure rates are needed to quantify the reliability model but are beyond the scope of this study. As noted previously, a failure rate of 10^{-08} per hour was selected as a placeholder for this application. Two software failure modes are defined for the application software on the CPUs and the controllers: software halt and erroneous output of software (each assumed to be 50% of the total failure rate).

Failure Mode/Mechanism Distributions of the Reliability Analysis Center [RAC 1997]

This document gives the failure modes of several components, including specific digital ones. Although the document contains failure modes distributions for almost all of the components of the DFWCS, only some of them are adopted in this study. The main difficulties in using these distributions are that (1) many failure modes provided seem like failure causes rather than failure modes and (2) many failure modes are difficult to understand as they lack an explanation. Therefore, it is often impossible to determine the failure effects on the components of the defined failure modes.

Electronic Components Selection and Application Guidelines [Meeldijk 1996]

This document is considered a supplementary reference to RAC [1997]. The failure modes provided in Meeldijk [1996] are more generic and less specific than those in RAC [1997], although easier to understand and use for failure effect analysis.

The way failure modes distributions are modeled in this study is that distributions for a specific component in [RAC 1997] are used in the analyses if they are applicable, understandable, and complete (i.e., the probabilities sum to unity). Otherwise, failure modes distributions from Meeldijk [1996] are used, if available. The last source for failure mode distributions is Aeroflex [2005], which only presents failure data and failure modes for MUX/DEMUX. As mentioned in the previous section, the failure rates from Aeroflex [2005] were used in this study because they are provided for different MUX/DEMUX failure modes.

The failure modes distributions of the major components of DFWCS modules are summarized here.

1. Microprocessor

   The failure modes “wrong data word” accounts for 60% of the total failures, and “processor stops updating output” accounts for the remainder [RAC 1997]. The failure modes distribution from RAC [1997] appears to be more specific than the other sources for processors.

2. Associated components of microprocessors, such as the Industry Standard Architecture (ISA) bus, RAM, ROM, Basic Input/Output System (BIOS), flash disk, and buffer

   RAC [1997] failure modes for these components (e.g., the RAM failure modes such as electrical failure, shorted, and contamination) are typical examples of failure modes that are difficult to apply in the analysis (i.e., the failure impacts cannot be determined from them). The same issue exists for the information in Meeldijk [1996]. Therefore, it was conservatively assumed that there is only one failure mode for each component, i.e., a loss of the component, which means the loss of all functions it performs. Then, the impacts of these component failures on the associated CPU or controller were postulated based on a general understanding of digital systems.

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
A/D Converter	OORH (Out of Range Hi)	$2.4X10^{-09}$	5	4	0-8	$9.6X10^{-11}$	All analog input signals	1. Failure rate of 16-bit A/D converters in PRISM [RAC Manual] using RAC Rates model; 2. Failure distribution of linear ICs in [Meeldijk 1996]
	OORL (Out of Range Low)			44	22-66	$1.1X10^{-09}$		
	Random Bit Failure			52	100- above	$1.3X10^{-09}$		
Address Logic	Loss	$7.0X10^{-08}$	16	100		$7.0X10^{-08}$	Address signals used to locate devices	1. HBM updated failure rate of decoder
Voltage Regulator	OORH	$3.7X10^{-09}$	5	50	25-75	$1.9X10^{-09}$	All analog voltage signals	1. Failure rate of voltage regulator in PRISM using RAC Rates model; 2. Failure mode distribution is assumed to be 50% each for OORH and OORL.
	OORL			50	100- above	$1.9X10^{-09}$		
Current Loop	OORH	$2.4X10^{-09}$	5	2	0-4	$4.8X10^{-11}$	All analog current I/O signals	1. Failure rate of IC, linear transmitter/receiver, a major component of current loop, in PRISM using RAC Rates model; 2. Failure distribution of linear ICs in [Meeldijk 1996]
	OORL			44	22-66	$1.1X10^{-09}$		
	DftH (Drift High)			27	0.5*(100-above modes)	$6.5X10^{-10}$		
	DftL (Drift Low)			27	Same as above	$6.5X10^{-10}$		
ROM	Loss	$4.0X10^{-08}$	14	100		$4.0X10^{-08}$	BIOS	1. HBM updated failure rate of ROM

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
Buffer	Loss	$3.9X10^{-07}$	88	100		$3.9X10^{-07}$	Digital Input/Digital Output (I/O)	1. HBM updated failure rate of buffer
Clock	Loss	$5.2X10^{-07}$	5	100		$5.2X10^{-07}$	ASIC will cease all functions upon the loss of clock reference.	1. Failure rate of clock generator in PRISM using RAC Rates model
D/A Converter	OORH	$2.4X10^{-09}$	5	2	0-4	$4.8X10^{-11}$	All analog output signals	1. Failure rate of 16-bit D/A converters in PRISM using RAC Rates model; 2. Failure distribution of linear ICs in [Meeldijk 1996]
	OORL			44	22-66	$1.1X10^{-09}$		
	DftH			27	0.5*(100-above modes)	$6.5X10^{-10}$		
	DftL			27	Same as above	$6.5X10^{-10}$		
Solid-State Switch	NCFC (Normally Closed, Fails Closed)	$2.43X10^{-09}$	5	67	100-below	$1.6X10^{-09}$	All digital I/O signals	1. Failure rate of solid-state switch in PRISM using RAC Rates model; 2. Failure distribution of solid-state switch in [RAC 1997]
	NCFO (Normally Closed, Fails Open)			33	0-67	$8.1X10^{-10}$		
	NOFC (Normally Open, Fails Closed)	$2.43X10^{-09}$	5	33	0-67	$8.1X10^{-10}$		
	NOFO (Normally Open, Fails Open)			67	100-above	$1.6X10^{-09}$		

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
DEMUX	Loss of All Signals	$8.8X10^{-09}$	5	100		$8.8X10^{-09}$	Analog output signals	1. Failure rates for the failure modes of a DEMUX are given in [Aeroflex 2005]. Therefore, the failure modes distribution is not needed.
	Loss of One Signal	$1.1X10^{-07}$	5	100		$1.1X10^{-07}$		
Flash Disk	Loss of Flash Disk	$3.3X10^{-07}$	76	100		$3.3X10^{-07}$	Storage of software and data	1. Same as the RAM data. Flash disk is actually RAM.
ISA Bus	Loss of ISA Bus	Sum of $4.6X10^{-07}$ and $6.2X10^{-08}$	55 and 10	100		$5.2X10^{-07}$	I/O bus between microprocessor and peripheral devices	1. HBM updated failure rate of line/bus driver and receiver
MUX	Loss of All Signals	$8.8X10^{-09}$	5	100		$8.8X10^{-09}$	Analog input signals	1. Failure rates for the failure modes of a MUX are given in [Aeroflex 2005]. Therefore, the failure mode distribution is not needed.
	Loss of One Signal	$1.1X10^{-07}$	5	100		$1.1X10^{-07}$		
Programmable Array Logic (PAL)	Loss of PAL	$1.6X10^{-09}$	5	100		1.6R-09	Failure of PAL will cause some user-written F-TRAN software to fail to run.	1. Failure rate of PAL in PRISM using RAC Rates model
RAM	Loss of RAM	$3.3X10^{-07}$	76	100		$3.3X10^{-07}$	Loading software to be executed	1. HBM updated failure rate of RAM
Software	Halt	$1X10^{-08}$	NA	50		$5.0X10^{-09}$	Performing functions of the system	1. Both failure rate and failure mode distribution are assumed.
	Error			50		$5.0X10^{-09}$		

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
Microprocessor	Error	$3.3X10^{-08}$	16	60	100-below	$2.0X10^{-08}$	Executing software	1. HBM updated failure rate of microprocessor
	Stop Updating			40	0-80	$1.3X10^{-08}$		
Flux Sensor	OORH	$5.0X10^{-06}$	5	40	0-80	$2X10^{-06}$	Flux measurements	1. [SRS 1993] (also, in Table 8-12 of NUREG/6962) 2. Failure modes distribution for "Sensor, Radiation" in [RAC 1997].
	OORL			60	100-above	$3X10_{-06}$		
Level Sensor	OORH	$5.0X10^{-07}$	3	42.8	0-85.6	$2.1X10^{-07}$	Level measurements	1. [SRS 1993] (also, in Table 8-12 of NUREG/6962 [Chu 2008a]) 2. Failure mode distribution for "Sensor, Level" in [RAC 1997].
	OORL			57.2	100-above	$2.9X10^{-07}$		
Flow Sensor	OORH	$3.0X10^{-06}$	3	31.5	0-63	$9.0X10^{-07}$	Flow measurements	1. [SRS 1993] (also, in Table 8-12 of NUREG/6962) 2. Failure mode distribution for "Sensor, Flow/Velocity" in [RAC 1997].
	OORL			68.5	100-above	$2.1X10^{-06}$		

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
Transmitter for Flux Sensor	OORH	$3.0X10^{-06}$	10	50	25-75	$1.5X10^{-06}$	All flux measurements	1. Failure rate in [SRS 1993] (also, in Table 8-12 of NUREG/6962) 2. Failure mode distribution is assumed to be 50% each for OORH and OORL
	OORL			50	100-above	$1.5X10^{-06}$		
Transmitter for Level Sensor	OORH	$3.0X10^{-06}$	10	20	0-40	$6.0X10^{-07}$	All level measurements	1. Failure rate in [SRS 1993] (also, in Table 8-12 of NUREG/6962) 2. Failure mode distribution for "Sensor, Level, Transmitter" in [RAC 1997]
	OORL			80	100-above	$2.4X10^{-06}$		
Transmitter for Flow Sensor	OORH	$3.0X10^{-06}$	10	45	0-90	$1.4X10^{-06}$	All flow measurements	1. Failure rate in [SRS 1993] (also in Table 8-12 of NUREG/6962) 2. Failure distribution for "Sensor, Flow/Velocity, Transmitter" in [RAC 1997]
	OORL			55	100-above	$1.7X10^{-06}$		
DC Power Supply	Loss	$1.0X10^{-05}$	10	100		$1.0X10^{-05}$	DC power to CPUs and controllers	1. [Wierman 2002]
AC Bus	Loss	$5.0X10^{-07}$	3.5	100		$5.0X10^{-07}$	AC power supply	1. [SRS 1993]
β -factor	N/A	0.05	3	N/A		N/A	CCF parameter	1. ALWR Utility Requirements Document [EPRI 1993]

Table 6-1 Failure data used in quantifying the DFWCS reliability model.

Components	Failure Modes	Overall Failure Rates		Filure Mode Distribution (%)		Failure Rates of Individual Failure Modes (per hour)	Related Signals or Fuctions	Data Sources
		Mean (per hour)	Error Factor	Mean	Uncertainty Bounds			
The data are used in this project to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes).								
CCF of CPUs	CCF	$1.5X10^{-05}$ *β(0.05)	N/A	100		$1.5X10^{-05}$	Main and backup CPUs	1. By adding failure rates of all CPU components including power supplies
CCF of Controllers	CCF	$2.7X10^{-06}$ *β(0.05)	N/A	100		$2.7X10^{-06}$	MFV and FWP controllers	1. By adding failure rates of all MFV controller components; 2. CCF of controller power supplies is modeled separately
CCF of Power Supplies of Controllers	CCF	$1.1X10^{-05}$ *β(0.05)	N/A			$1.1X10^{-05}$	MFV and FWP controllers	1. By adding failure rates of DC and AC power supplies

3. Address logic

This is a generic digital component, also called the decoder. The failure modes distribution in RAC [1997] for this component again is difficult to apply. Hence, the failure mode distribution of a typical digital component from Meeldijk [1996] was selected: stuck high (40%), stuck low (40%), and loss of logic (20%).

4. Voltage input module

The voltage regulator is assumed to be the major component of the voltage input module. RAC [1997] gives the failure modes and the distribution (50% each for fail-high and fail-low).

5. MUX and DEMUX

Aeroflex [2005] defines the failure modes. Note that each input of a MUX corresponds to a sensor input, and each output of a DEMUX corresponds to an analog output. Also, a loss of one signal and a loss of all signals are the only failure modes of a MUX or a DEMUX included in Aeroflex [2005].

6. A/D and D/A converters

Each module has only one A/D converter and one D/A converter; they are shared, respectively, by all analog inputs and analog outputs. Both A/D and D/A converters are linear integrated circuits (IC), i.e., the inputs to and outputs from the component are proportional to each other. The distributions of failure modes given in RAC [1997] for A/D or D/A converters again are difficult to use. Thus, the failure modes distribution defined in Meeldijk [1996] for a linear IC component was used: drifted output (52%; degraded/improper output [50%] and drift [2%]), fail-low (44%; no output [41%] and short circuit [3%]), and fail-high (2%; open circuit). For A/D converters, the failure mode distribution was obtained by assigning the D/A converter failure modes to A/D convert failure modes, i.e., fail-high and fail-low were assigned to all bits stuck at zeros and ones (48%)[(1)], and drifted output was assigned to random bit failure (52%).

7. Current input and output (I/O) modules (current loops)

Linear transmitter/receivers are the major component of current input modules and current loops. They also are linear IC devices; therefore, the failure mode distribution used was the same as for A/D and D/A converters above.

8. Voltage regulator

The failure modes distribution for voltage regulators was not found in available references. Therefore, it was assumed to be 50% for both OORH and OORL.

[(7)] The failure modes distributions for linear IC circuits in Meeldijk [1996] do not sum to 100%. For this study, the distribution of failure modes for A/D converters was modified by changing the percentage for the failure mode “all bits stuck at zeros or ones” from 46% to 48%. Due to an oversight, a similar modification was not made to the failure mode percentages for D/A converters.

9. Digital input and digital output modules

Digital input and digital output are implemented using a solid-state switch [Eurotherm 2000]. The status of the output is controlled by opening or closing this switch. Its failure modes distribution is fail to operate (fail as is) (66.7%) and false operation (fails to opposite state) (33.3%) [RAC 1997].

10. Sensors

RAC [1997] provides the failure mode distributions of different sensors. Because the assumed failure mode for sensors in this study were OORH and OORL, some of the failure modes provided in RAC [1997] needed to be further split, e.g., the failure mode "degraded output" was assumed to ultimately progress to either OORH or OORL, with equal likelihood, as indicated in parentheses below.

For flow sensors, the failure distribution of "Sensor, Flow/Velocity" from [RAC 1997] was used: degraded output (41.4%; OORH [assumed 50%] and OORL [assumed 50%]), zero or maximum output (21.6%; OORH [assumed 50%] and OORL [assumed 50%]), no output (OORL) (18.5%), function without signal (OORL) (14.8%), no operation (OORL) (2.5%), and cracked (OORL) (1.2%).

For level sensors, the failure distribution of "Sensor, Level" from [RAC 1997] was used: degraded output (54.7%; OORH [assumed 50%] and OORL [assumed 50%]), no output (OORL) (20.8%), function without signal (OORL) (14.2%), and zero or maximum output (10.4%; OORH [assumed 50%] and OORL [assumed 50%]).

For flux sensors, the failure distribution of "Sensor, Radiation" from [RAC 1997] was used: degraded output (53.3%; OORH [assumed 50%] and OORL [assumed 50%]), zero or maximum output (26.7%; OORH [assumed 50%] and OORL [assumed 50%]), and no output (OORL) (20%).

11. Transmitters

Failure modes distributions of transmitters of different sensors also were obtained from RAC [1997]. Their assumed failure modes were OORH and OORL. Therefore, some failure modes were further split, as described above for sensors. For the transmitter of a flow sensor, the failure distribution of "Sensor, Flow/Velocity, Transmitter" from [RAC 1997] was used: degraded output (50%; OORH [assumed 50%] and OORL [assumed 50%]), zero or maximum output (40%; OORH [assumed 50%] and OORL [assumed 50%]), and no operation (OORL) (10%). For the transmitter of a level sensor, the failure distribution of "Sensor, Level, Transmitter" from [RAC 1997] was used: degraded output (47.1%; OORH [assumed 50%] and OORL [assumed 50%]), zero or maximum output (35.3%; OORH [assumed 50%] and OORL [assumed 50%]), and no output (OORL) (17.6%). There are no data on failure distribution for the transmitter of a flux sensor. Thus, it was assumed that the OORH and OORL each are 50% of total failures.

6.3 Common-Cause Failures (CCFs)

The DFWCS consists of two identical CPUs that run identical software. In this study, the system failure of the DFWCS is defined as a loss of automatic control, implying that a CCF of either the CPU hardware or software will cause a system failure. Controllers may also experience CCFs since they have identical hardware and similar software, as do the sensors and transmitters that have redundancy. Therefore, CCF data on these modules and components are needed in the quantification.

Due to the lack of digital-specific CCF parameters and because developing a database for CCF parameters of digital components is beyond the scope of this project, it was decided that the generic beta factor suggested in the Advanced Light Water Reactor (ALWR) Utility Requirement Document (URD) [Electric Power Research Institute (EPRI) 1993], i.e., 0.05, be used. The ALWR URD does not specifically address digital components and suggests using the generic CCF parameter for components whose specific parameters are unavailable.

The use of the beta factor in developing CCF failure rates is summarized below.

CPU Modules

In this proof-of-concept study, the CCF of the CPU modules is treated as a failure of a "pseudo-component" that contains all of the major components of a CPU module. The failure rate of the CCF was calculated by adding the failure rates of the failure modes of all components contained in the "pseudo-component" and multiplying the sum by a beta factor. It is assumed that the CCF causes system failure, which is conservative because not all of the failure modes included in the pseudo-component cause system failure. In a more realistic application, the failure modes that fail and do not fail the system should be modeled separately.

Controller Modules

The CCF of the controllers is modeled in the same way as is the CCF of the CPU modules.

Sensors and Transmitters

The CCFs of each type of sensor and associated transmitter are quantified using the beta factor of 0.05. Note that not every such CCF would lead to system failure, and the failure effects as determined by the FMEA are reflected in the reliability model. For example, the CCF of steam flow sensors will switch the control from 3-element to 1-element control, and additional failures will have to take place to result in a system failure.

120v AC Buses and DC Power Supplies

The CCFs of the 120v AC buses and DC power supplies that support the controller modules are quantified using the beta factor of 0.05; each will cause the system to fail. As discussed in Section 3.3.6, the CCFs of the 120v AC buses and DC power supplies that support the CPU modules are quantified similarly, and have the same effect on the system, but for simplicity are modeled by including their contribution in the CCF of the CPU modules, not as a separate failure mode.

The effect of adverse operating environment on digital components should be accounted for in the quantification of the CCF beta factors. In particular, electromagnetic interference (EMI) is a unique phenomenon which may affect operation of digital systems by altering the signals processed by the systems. In general, the digital system design should take into consideration its operating environment to protect against EMI. Only unexpected EMI or failure of the protection mechanisms, e.g., shielding, should cause adverse EMI effects on the system. One possible way of accounting for the adverse effect of EMI is assuming that it would cause a system failure and modeling it as a single failure like a CCF. The failure rate of the single failure can be estimated by analyzing the potential causes, i.e., sources of unexpected EMI and failure of the protection mechanism. This type of analysis is beyond the scope of this study.

7. QUANTIFICATION

In this study, a Markov model of the digital feedwater control system (DFWCS) was developed and quantified. In general, the possible system states are defined in terms of the states of its components; hence, the number of system states grows exponentially with the number of components and may become unmanageable for detailed models, such as the model developed in this study. The issue of this explosion of states was addressed by truncating system failure sequences based on their order (i.e., the number of individual failures included in the sequence) and demonstrating that convergence of system failure probability is achieved. This is consistent with the understanding that the sequences with a larger number of failures needed to cause system failure tend to have lower probabilities. The Markov model was solved considering only those sequences with three or fewer failures, while estimating an upper bound error of truncation to demonstrate convergence of the system failure probability. Section 7.1 presents the results of the quantification of the Markov model. Section 7.2 presents an approximate method for quantifying the sequences, i.e., a quantification method using the rare event approximation (i.e., the simplified Markov model presented in Section 5.3). Section 7.3 provides a comparison of the results with the operating experience of the system. It should be pointed out that the failure parameters used in the quantification are weak, and it was decided not to include a list of dominant sequences in this report.

Uncertainty analysis and sensitivity analyses are discussed in Chapter 8.

7.1 Quantification of Markov Model

Chapter 5 discusses the Markov model of the DFWCS, including the analytical solution to the model. Equations (5-4) to (5-6) are used to quantify the sequences of failure modes identified in Chapters 3 and 4 that cause system failure. Table 7-1 summarizes this quantification. There are 112 single failures; 39,497 double-failure sequences; and 11,972,960 triple-failure sequences. The table also lists the probabilities of those sequences that do not cause system failure. They represent the maximum that may be missed in calculating the probability of system failure if the quantification is stopped at the respective numbers of failure modes (i.e., sequence order). For example, if the quantification is stopped at sequences with only one failure, i.e., the Markov transition diagram does not expand beyond layer 2 in Figure 5-2, there are 309 individual failures that do not cause system failure and they have a total probability of 0.47. With the quantification stopped at sequences with 3 failures, the upper bound of the error due to truncation becomes 0.02, demonstrating the decreasing trend of the error. This will be further illustrated below by using Figures 7-1 to 7-3.

It is an advantage of the method that the error of truncation can be estimated analytically which can be used in determining if convergence has been achieved. The last column shows the cumulative probabilities of system failure obtained by successively adding the contributions of the single failure modes, double sequences, and triple sequences. The contribution of single failures is the highest, followed by that of double failure sequences; the contribution from the triple failure sequences is only a small fraction of the total probability. The cumulative probabilities shown in the last column of Table 7-1 indicate that the total system failure probability is converging and should be fairly close to the actual system failure probability.

Table 7-1 Quantification of system failure probability and frequency.

	Number of sequences that cause system failure	Number of sequences that do not cause system failure	Probability of sequences with system failure	Probability of sequences without system failure	Total system failure probability (frequency per year)
Individual failure modes	112	309	0.051	0.47	0.051 (0.052)
Sequences of two failure modes	39,497	89,282	0.023	0.12	0.074 (0.077)
Sequences of three failure modes	11,972,960	24,871,719	0.0052	0.02	0.079 (0.083)

In general, higher order failure sequences could be generated and quantified to produce more accurate results. However, in this proof-of-concept study, quantification was stopped at the triple-failure level, since this is considered adequate to demonstrate the trend of convergence.

To better understand the results shown in Table 7-1 and the convergence criteria used for stopping expansion of the Markov transition diagram, numerical values that represent probabilities with and without system failure are labeled in Figures 7-1 to 7-3 if we stop expanding the Markov transition diagram at individual, double, and triple sequences, respectively. Note, the probability of being in the perfect state is always the same.

Using the total system failure probability (0.079, based on all failure paths with three or less component failures) in Equation (1-1), i.e., $f = -\ln[R(T)]/T$, the frequency of loss of automatic control of the DFWCS is calculated to be 0.083 per year[8] .

Some digital instrumentation and control system models may require that a very large number of sequences be quantified using the Markov method. The computational effort required may be tremendous, especially if one has to integrate multiple, interactive digital systems in the analysis. However, it should be recognized that the proposed approach inherently is capable of parallel processing because quantification of the sequences are not related to each other and can be processed independently. Therefore, a linear scalability of the quantification can be achieved by distributing the sequences onto multiple computers, and the results can be collected and combined. This offers a practical solution for the complexity and scale of digital systems.

[8] In this study, the time period T used in the quantification was one year. It is more appropriate to use the refueling cycle length (18 months) as T, assuming that the system will be renewed every refueling.

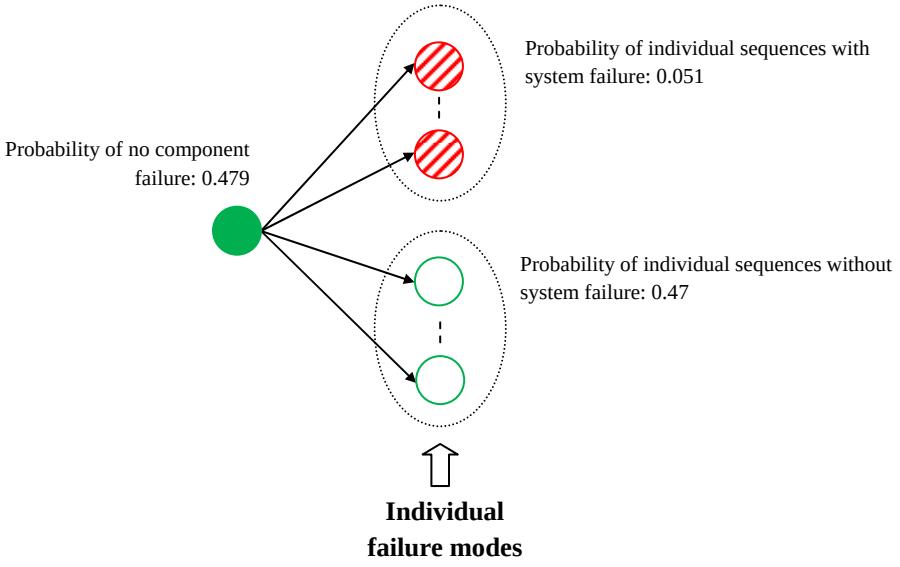

Figure 7-1 Markov diagram for and quantification of individual failure modes

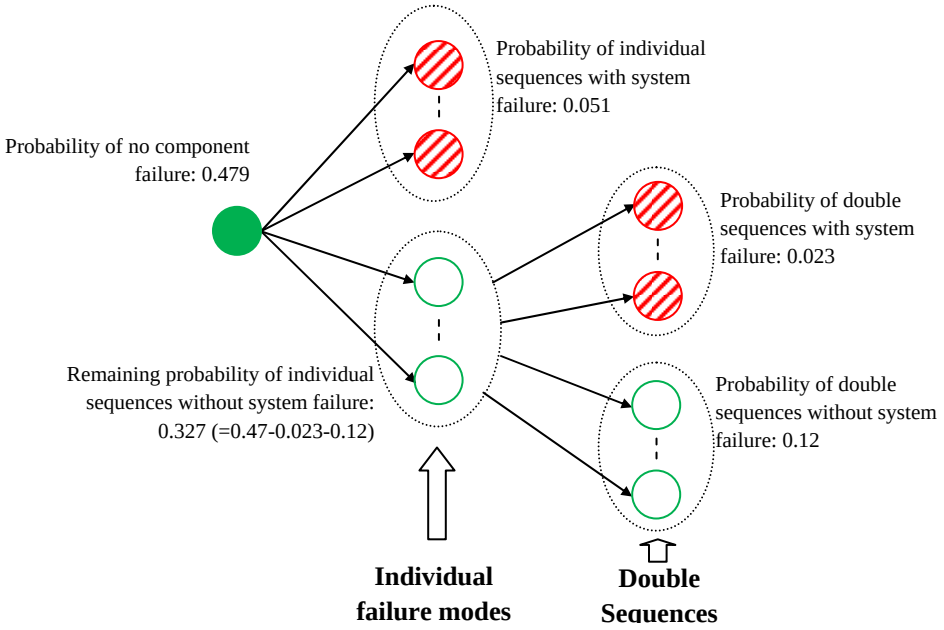

Figure 7-2 Markov diagram for and quantification of both individual and double sequences

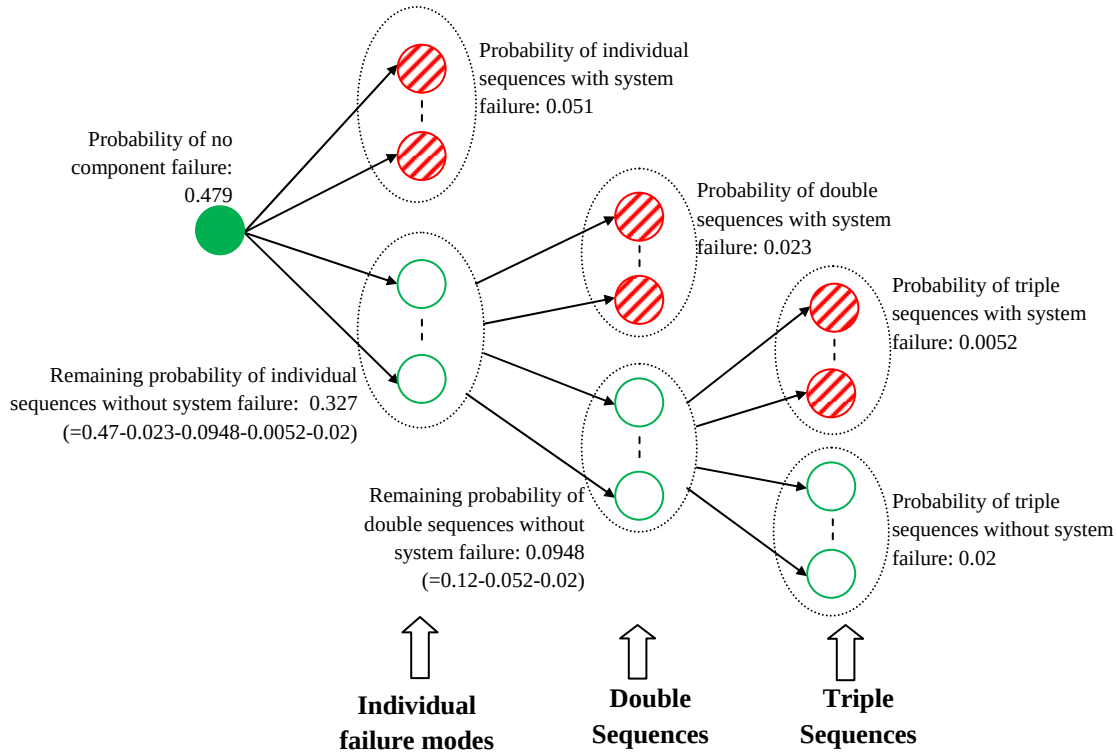

Figure 7-3 Markov diagram for and quantification of individual, double, and triple sequences

7.2 Approximate Quantification of Markov Model

The failure sequences of the DFWCS were also quantified using an approximate quantification method, the rare event approximation described in Section 5.3. Since the frequency of failure for the DFWCS is dominated by single failures, the deviation from the exact result may not be as large as might be expected for a system that involves a greater degree of redundancy, e.g., a reactor protection system. Table 7-2 summarizes the frequency of loss of automatic control calculated using the exact and simplified methods.

Table 7-2 Frequency of loss of automatic control.

	Exact Method	Simplified Markov Model
Frequency of Loss of Automatic Control (per year)	0.083	0.12

Using the simplified Markov method, the probability of a component failing to one of the failure modes is calculated in a way that implies that the component has only one failure mode. In doing so, the competition between different failure modes is ignored, i.e., the non-occurrence of all other failure modes is not accounted for. The results of the table show that the frequency of loss of automatic control has a point estimate frequency of 0.083 per year, indicating that such an initiating event is not very rare. Due to this fact, the rare event approximation method (i.e., the simplified Markov method) only provides a rough estimate of the frequency.

7.3 Comparison with Operating Experience

In the plant for which the DFWCS of this study was primarily based on, the DFWCS initially was installed and first used in the early 1990s. Over the years, the system apparently has undergone significant modifications. Two failure events have been reported related to the system. In one, a maintenance-induced loss of 120v alternating current (AC) power to the main and backup CPUs, combined with the independent failure of the main feedwater-regulating valve positioner-selector solenoid valve, caused a low steam-generator level and automatic plant trip. This study models loss of power supply, i.e., 120v AC, which is an important contributor to the top event. The positioners are beyond the scope of this study. In the second event, the main feedwater valve (MFV) controller generated a slowly increasing signal, leading to an increase in steam-generator level and the reactor was manually tripped. The cause of the failure was most likely electromagnetic interference due to inadequately shielded cables. The MFV controller generating a drifting demand signal is a failure mode of the system that the model used in this study does not explicitly consider; however, arguably, it is covered by one of the failure modes of the MFV controller, e.g., the MFV controller microprocessor. One complete loss of feedwater control (from modeled failure modes) in 30 reactor-years of operating experience is not inconsistent with the estimated mean frequency for loss of automatic feedwater control of 0.08 per year for one DFWCS. Note, there are a few reasons the results of this study may not be suitable to compare with operating experience: (1) the model does not take into consideration that manual control may be possible for some of the failure sequences, and (2) as stated in Chapter 6, the failure parameters used in this study have very large uncertainty.

8. UNCERTAINTY ANALYSIS AND SENSITIVITY CALCULATIONS

In a probabilistic risk assessment (PRA), three types of uncertainties need to be considered: parameter, model, and completeness uncertainty. Typically, parameter uncertainty is addressed by assessing the uncertainties associated with the parameters used in the model and propagating them through the PRA model. Modeling uncertainty is addressed by identifying modeling assumptions and performing sensitivity calculations to evaluate the impacts of alternative assumptions and modeling methods on the results. Incomplete PRA scope or incomplete PRA level of detail can be addressed by using screening or bounding analyses to demonstrate that the missing items are not risk significant.

In this study, parameter uncertainty is addressed in the typical way, as documented in Section 8.1. Modeling uncertainty analysis is addressed by documenting the assumptions made in developing the model and comparing with models developed using dynamic methods. Due to limitations in the state-of-the-art and lack of detailed design information, no sensitivity calculations were performed to evaluate the effects of alternative assumptions. Section 8.2 provides a summary of the assumptions and possible ways of addressing them. Chapter 10 is a high-level comparison of the model developed in this study with those of the dynamic methods [Aldemir 2009], and can be considered a way of addressing some aspects of modeling uncertainty. Completeness uncertainty was dealt with in a limited way by identifying sources of incompleteness in the probabilistic model developed, and they are briefly discussed in Section 8.3.

The simplifications and assumptions made in this study can be characterized as addressing either (1) scope and level of detail limitations or (2) state-of-the-art limitations. Scope and level of detail limitations represent those that can be removed by expanding the scope of the study and increasing the level of detail of the model. These limitations are not inherent limitations of the method developed in this study. State-of-the-art limitations represent weaknesses in the state of the art and might be resolved by performing additional research. Sensitivity calculations can be used to demonstrate the importance of the needed research. The simplifications and assumptions associated with each group of limitations are identified in Sections 8.2 and 8.3.

Regarding software reliability, this study accounts for the normal behavior of software by using a simulation tool that runs the actual software of the system to determine the system response to postulated hardware failures. This study also includes placeholders for software failure events, assuming that the basis for modeling software failure in this way can be established. It remains to be seen whether such a basis can be established and accepted by the PRA community. In addition, methods for quantifying software failure rates and probabilities have to be developed (in this study, arbitrarily chosen failure rates are included for the placeholder software failure events). Quantification of software reliability is a limitation in the current state of the art.

The treatment of software reliability in this study contributes to all three types of uncertainty. There is parameter uncertainty because the values of the failure rates for these failures have wide variability, model uncertainty because there is currently no widely accepted model for quantifying and modeling these failures, and completeness uncertainty because it is not known whether the scope and level of detail of these failures is appropriate, and not all possible failures have been modeled.

In order to demonstrate the usefulness of the reliability model developed in this study, Section 8.4 documents some calculations that were performed to evaluate the importance of

selected digital design features of the digital feedwater control system (DFWCS). The digital features of interest include redundancy in the central processing unit (CPU), the external watchdog timers (WDTs), the controller demand feedback to CPUs, and out-of-range (OOR) checking of analog signals.

8.1 Parameter Uncertainty

The failure parameters needed to quantify the reliability model of the DFWCS include the failure rates of individual components, the distribution of failure modes of each component failure, and a beta factor that is used to model the common-cause failures (CCFs) of components or modules. Table 6-1 lists these data, including their uncertainty parameters. Chapter 6 provides additional information on the estimation of failure parameters, and NUREG/CR-6962 [Chu 2008a] documents details of the data analyses. Due to recognized weaknesses in the data, the data values are used in this project solely to demonstrate the reliability methods and exercise the reliability models. They are not appropriate for quantifying models that are to be used in support of decision-making (e.g., regulatory decisions or design changes). In this chapter, an uncertainty analysis is undertaken by propagating the parameter uncertainties shown in Table 6-1. In this analysis, the state-of-knowledge-correlation (SOKC) has to be accounted for since it may significantly affect the final result as shown by Apostolakis [1981] and Chu [2008b].

The parameter uncertainties can be propagated by sampling the distributions of the parameters and using them in quantifying the system failure probability. The following distributions are assumed for failure parameters:

1. The failure rate of a component is lognormally distributed;
2. The probability of occurrence of a failure mode in a failure-mode distribution is uniform; and
3. The beta factor is assumed to be lognormally distributed.

Parameters that characterize the distributions, e.g., the mean value and error factor for a lognormal distribution, and the upper bound and lower bound for a uniform distribution of a failure-mode probability in a failure-mode distribution, are used to generate samples of the failure rate of individual failure modes. Selection of the uncertainty bound for the failure-mode distributions in Table 6-1 guarantees that the sum of the samples of failure-mode probabilities of a failure-mode distribution is 1.0. In each quantification step, the failure rate of each component failure mode is obtained by multiplying the sample of a component failure rate by a sample of the probability of the corresponding failure mode of the failure-mode distribution. CCF rates are calculated as the products of samples of the beta factor and the component failure rates.

To reduce the number of CCFs in the model, the CCFs of the CPU and controller modules each are represented by a single failure event whose rate is determined by multiplying the lumped failure rate of all components in a CPU or a controller module by a beta factor. It is noted that CCF of the CPU power supplies is included as a part of the CPU module CCF, but the CCF of the controller power supplies are modeled separately because they are depended upon by all four controllers.

As mentioned previously, in accounting for the propagation of the uncertainties in the parameters, an issue to consider is the SOKC. Chu [2008b] demonstrated the impacts of the

SOKC on top events of a PRA showing that that the impacts of correlation are significant if the number of identical components and/or the error factor in the distribution of component failure rate is relatively large.

The DFWCS employs several identical components in its modules, e.g., current loops are used for analog inputs and outputs (I/O), and solid-state switches are used for digital I/O as discussed in Chapter 3. In this study, to take the SOKC into account, those components using the same failure parameters are considered correlated; thus, all current loops or solid-state switches of different modules are deemed to be correlated. In addition, the same beta factor with a mean value of 0.05 is used to model all CCFs, thus, the beta factors used for different CCFs also are correlated.

To account for the SOKC, a large number of sets of samples should be generated from the parameter distributions of Table 6-1, using each once in quantifying the probability of the top-event (in this case, the system failure). The resulting samples of system-failure probabilities then are used to estimate the statistical characteristics of the system-failure probability or frequency, such as the mean, median, and 5th and 95th percentiles. For example, a single sample taken from the failure-rate distribution of a current loop is employed for all current loops of the DFWCS model for that particular sample calculation. Similarly, a sample taken from a failure-mode distribution becomes the failure-mode distribution of all components that share the same such distribution. In each step of the quantification, a failure rate for each component failure mode is obtained by multiplying a sample of the component failure rate by a sample of the corresponding failure-mode probability taken from the failure-mode distribution. In the same way, samples of beta factor of CCFs are generated and used, i.e., a sample of the CCF rate is obtained by multiplying a failure-rate sample of a single component or a lumped failure-rate sample of a module by a sample of the beta factor.

Due to time constraints in this study, only 1000 samples were generated and used as input to the quantification. The mean value of the calculated system-failure probabilities is 0.067, i.e., smaller than the point-estimate system-failure probability of the base case (0.079). The 5th, 50th, and 95th percentiles of these probabilities are 0.012, 0.032, and 0.23, respectively. More accurate results are expected with more samples.

Table 8-1 summarizes the results of an uncertainty analysis of the frequency of loss of automatic control. The mean frequency is 0.069 per year, with an error factor (square root of the ratio of the 95th and 5th percentiles) of 4.7. Note that the frequency is related to the system failure probability, P(T), by the equation $frequency = -\ln[1 - P(T)]/T$, where T is one year.

Table 8-1 Results of uncertainty analysis for frequency of loss of automatic feedwater control (per year).

5th Percentile	Median	95th Percentile	Mean	Point Estimate
0.012	0.033	0.26	0.069	0.079

8.2 Modeling Uncertainty

As mentioned previously, modeling uncertainty is typically addressed by identifying modeling assumptions and performing sensitivity calculations to evaluate the impacts of alternative assumptions and modeling methods on the results. In this study, the treatment of modeling uncertainty was limited to documenting the assumptions made in developing the model and comparing with models developed using dynamic methods. Due to limitations in the state of the art and lack of detailed design information, no sensitivity calculations were performed to evaluate the effects of alternative assumptions.

Many of the assumptions made in this study contribute primarily to completeness uncertainty and are addressed in Section 8.3. A few modeling assumptions are listed below with references provided to earlier chapters where more discussion is available.

- Typically, a component has more than one failure mode. A component is assumed to fail only once in a given failure sequence, i.e., after one failure mode of the component has occurred, other modes cannot occur for the same component. Section 3.3 provides more discussion of this assumption.

- Due to lack of detailed design information, failures of different components are assumed to be independent of each other (regardless of how they are physically wired together). Section 3.3 provides more discussion of this assumption.

- Components of the system cannot be repaired or replaced while the system is operating. Section 5.4 discusses how repair can be accounted for in a Markov model, e.g., using the simplified Markov model as a quantification method.

- Due to state of the art, some assumptions are made regarding the failure modes, failure effects and failure detectability of components. They are discussed in Chapter 3 and Appendix A.

- Due to lack of detail design information, assumptions are made regarding the arrangements of alternating current and direct current power supplies to the system. Section 2.6 provides more information on these assumptions.

8.3 Completeness Uncertainty

Completeness uncertainty relates to contributions to risk that have been excluded from the PRA model. Lack of completeness is not in itself an uncertainty, but recognition that some risk contributors may be missing from the PRA model. The result is, however, an uncertainty about where the true risk lies. In this study, completeness uncertainty was dealt with in a limited way by identifying sources of incompleteness in the probabilistic model developed, as follows:

- Lack of a thermal-hydraulic plant model that interfaces with the DFWCS − by definition, this study uses “traditional” methods and does not explicitly model the plant physical processes. As discussed in Sections 3.4 and 4.4, drifting signals are difficult to model with or without a plant model. The contribution of this failure mode to system failure can be captured by conservatively assuming that system failure would result. The sensitivity

calculation on OOR checking and deviation logic discussed in Section 8.4 provides a way of estimating the importance of drifting signals.

- Lack of modeling of manual control of feedwater − This study assumes the plant is operating at full power, and models the loss of automatic control of feedwater, i.e., no consideration is given to the possibility of manual control. In general, it is possible to examine the individual failure sequences to determine if feedwater could be manually controlled using the valve positioner and pump turbine speed controller.

- Lack of consideration of the impact of adverse environments on the digital systems – This study assumes that the system is not subject to adverse environments, e.g., loss of heating, ventilating, and air conditioning (HVAC) and exposure to electromagnetic interference (EMI), because the system is inside the control room which has good room cooling and should be protected against EMI. Impact of loss of HVAC is considered insignificant as discussed in Section 2.6. Section 6.3 indicates that EMI impact can be modeled by evaluating sources of unexpected EMI and failure of the protection mechanisms.

- Lack of a detailed model of connected digital systems − It is recognized that the main feedwater valve (MFV) positioners and turbine controllers are also digital systems. Due to lack of design information, they were considered beyond the scope of this study. In general, the method of this study can be applied to model these digital systems.

- Modeling function of external watchdog timers (WDTs) only − The external WDTs monitor the toggling signal from a digital output of the main or backup CPU and send out the status signal (digital) of the main or backup CPU to the MFV, bypass feedwater valve (BFV), and feedwater pump (FWP) controllers. In this study, the functions of the WDTs are considered (e.g., identification of the WDT-detectable failures) while the failure modes of the WDTs are not modeled due to a lack of design information of the WDTs. The failure modes of WDTs, which could be either a failure to indicate the failure status of the associated CPU when the CPU is failed or a spurious signal output indicating that the CPU is failed when it is not, can be accounted for by including failure modes of the WDT components that may contribute to the two failure modes in the reliability model.

- Simplified model for BFV and pressure differential indicating (PDI) controllers − As discussed in Section 3.3.5, the BFV controller automatic/manual status signal can cause a system failure and the PDI controller may inadvertently take over the control normally performed by the MFV controller. Both failure failure modes are included in the DFWCS model, and no other failure modes of the BFV and PDI controllers are included. In general, other failure modes of the controllers may contribute to the two failure modes that are included in the model, and the method of this study can be used to model them.

- Identification of failure sequences with more than three failures − The approach developed in this study addresses the state explosion issue by limiting the number of independent failures assumed in the failure sequences while demonstrating that convergence has been reached. This is a concept similar to that of cutset truncation typically done in a PRA. It is likely that more efficient generic software algorithms and tools can be developed to facilitate generation and quantification of higher order sequences.

- Failure modes − This study uses publicly available generic hardware failure modes of the components of the DFWCS and includes high-level failure modes of the software. It is recognized that some of the failure modes may not be complete and associated failure mode distributions may not be accurate. Software failure modes that are appropriate for inclusion in a reliability model also need to be established. Since software failures are beyond the scope of this study, the model only includes a couple of placeholder events for software failure (see Section 3.3.1). An example software failure mode that is not modeled is a failed output that is outside the acceptable range and detectable. This raises the question as to whether or not failure modes should be defined in terms of individual output signals of the CPU. It is clear that given the number of output signals associated with a CPU, this would result in a very large number of software failure modes, a number that would quickly become unmanageable when considering combinations of these output signals. Also, as stated in Section 3.3.1, the failure modes of the microprocessors of CPU modules considered in this study are at the same level of detail as those assumed for software (i.e., they are also not defined in terms of specific output signals of the CPUs). In the case of a failed output that is outside the acceptable range and detectable, the failure mode would likely be detected by the feedback signal from the MFV to the CPU. The failure modes considered in this study represent higher level failure modes whose completeness should be further examined as suggested in Chapter 11. The lack of completeness in identifying digital system component failure modes is a limitation in the current state of the art.

- Failure parameter database − This study uses publicly available component failure data at the level of detail of the model, and for some components performed an Hierarchical Bayesian Method analysis to account for variability of data from diverse sources. In the case of CCFs, practically no data is publicly available. It is recognized that better parameter data are needed in order to have confidence in the quantitative results. The lack of applicable failure parameter data is a limitation in the current state of the art.

Some other sources of completeness uncertainty include the following items, all of which are out of the scope of this proof-of-concept study: human reliability analysis associated with digital systems and human system interfaces (including indication errors), modes of operation other than full power, and software reliability.

8.4 Sensitivity Calculations

A few sensitivity analyses were carried out in this study to assess the benefits of different design features on the DFWCS reliability. The following design features were selected because of their potential impact on the DFWCS reliability:

1. A backup CPU that becomes the controlling CPU during the occurrence of a main CPU failover;

2. A WDT of the main/backup CPU that triggers a failover given the occurrence of certain failures;

3. Demand feedback signals from controllers to CPUs that are used to detect deviations between the CPU calculated demands and the controller demand outputs. A failover of the main CPU will occur when the deviations are large; and

4. Deviation logic of the CPUs.

To assess the benefits of each design feature, a sensitivity analysis was performed assuming the feature is unavailable, and the resulting system failure probability was compared to that of the base case.

8.4.1 Benefit of Redundancy in CPU

The DFWCS system benefits significantly from having redundant CPUs, as is evident from the number of failover occurrences initiated either by software or hardware that are identified in the failure modes and effects analyses (FMEAs) of Appendix A. Many individual failures would become single failures were it not for the backup CPU. The sensitivity analysis performed here compares the system-failure probabilities with and without the backup CPU.

In the sensitivity calculation, all failures that initiate a failover were assumed to cause a system failure, and a new rule was created in the automated FMEA tool to capture such system failures. The new rule states that a system failure occurs whenever there is a failover request; this is a slightly conservative assumption because some failures that initiate failovers do not necessarily fail the system. For example, according to the main CPU FMEA in Appendix A, a failed out-of-range high (OORH) or failed out-of-range low (OORL) of the analog input signal steam generator (S/G) 11 Level #1 of the main CPU will be detected, whereupon the other input S/G-11 Level #2 will be used for the control. A failover to the backup CPU will occur after a delay if the backup CPU is available. However, should the backup CPU be unavailable, the DFWCS will not fail if the other input S/G 11 Level #2 is valid.

The findings from the automated FMEA tool show that when there is no backup CPU, the total number of individual failures decreases from 421 to 290, but the number of such failures that directly result in system failure increases significantly from 112 to 170. Without the backup CPU, the failure probabilities of single, double, and triple sequences are 0.13, 0.024, and 0.0013, respectively. The total failure probability is around 0.15, yielding a loss of feedwater initiating frequency of 0.16 per year, compared with only 0.083 per year for the base case.

8.4.2 Effectiveness of Watchdog Timers

A WDT primarily monitors a digital output signal reflecting the status of a microprocessor. In the DFWCS, each of the main and backup CPUs has an external WDT. The FMEAs for the main and backup CPUs in Appendix A indicate that some failures are detectable by the WDTs and result in failovers. A sensitivity analysis was performed assuming that watchdog-detectable failures become single failures in the absence of the timer.

Similar to the backup CPU sensitivity analysis, a new rule was created in the automated FMEA tool to capture system failure and evaluate its probability assuming the WDTs are not available. It states that any WDT-initiated failover becomes a single failure.

Without the WDTs, the total failure probability of the system increases only slightly to 0.088 (corresponding to a frequency of 0.092 per year) from 0.079 in the base case. This result reflects the limited number of WDT-detectable failures.

8.4.3 Benefit of MFV Demand Feedback Signals

A specific design feature of the DFWCS is that the controllers send demand outputs back to the CPUs, as well as to the regulating valves and the pump. For example, the MFV controller sends demand output to the main feedwater-regulating valve as well as back to the CPUs. The CPUs compare the demand feedback to the calculated MFV demands. If the controlling CPU detects a large deviation between them, its application software will initiate a failover. If the main CPU calculates the MFV demand incorrectly due to internal failures of the main CPU module, such as a multiplexer failure, the backup CPU will take over and automatic control continues, which is an obvious benefit afforded by the deviation logic. However, if a large deviation is caused by certain failures of the MFV controller, the backup CPU also will be failed by the same deviation logic. It should be noted that the deviation logic will not capture a fail low of the MFV controller demand. Instead, it is considered to cause a system failure, captured by Rule 6 in Section 4.2.6, because the PDI controller that is not included in the reliability model will detect this failure first and take over (resulting in the need for manual system control).

Taking the MFV controller feedback as an example, a sensitivity analysis was performed by disabling the failover logic in the case of a large discrepancy between the demand feedback and the CPU calculated demand. It was assumed that the system fails when the MFV demand feedback deviates from the CPU calculated demand, e.g., a failed high of the demand feedback was assumed to cause a large deviation, initiating a failover.

Without the MFV deviation logic, the total DFWCS failure probability increases slightly from 0.079 to 0.080, corresponding to a loss of feedwater initiating frequency of 0.083 per year, essentially the same as that for the base case. This also suggests that the MFV demand-deviation logic is designed and effective only for a limited number of failures, i.e., MFV demand-related failures, and does not significantly improve the DFWCS reliability.

8.4.4 Benefit of Deviation Logic

This sensitivity analysis evaluates the benefit of the deviation logic of the CPU modules described in Section 2.1, i.e., (1) OOR and high rate of change (validity checks) for input signals, such as S/G level, feedwater flow and steam flow signals; (2) deviation checks, if redundant signals of the same type are valid; and (3) deviation checks for the controller demand feedback signals (for both the MFV and FWP controllers). Note that the latter is the same as the sensitivity analysis described in Section 8.4.3 except that both the MFV and FWP signals are considered. This sensitivity analysis is also related to that of the backup CPU since many OOR failures of analog signals will initiate a failover to the backup CPU.

For this sensitivity analysis, it was assumed that the deviation logic is disabled, i.e., there are no validity and deviation checks. Therefore, if a sensor input is OOR, it will lead to the control algorithms using incorrect values and generating incorrect output values. This was conservatively assumed to be a system failure. A rule was created that specifies a system failure when any of the following analog inputs to the controlling CPU is out of range: Feedwater Flow # 1 and #2, S/G 11 Level #1 and #2, Steam Flow #1 and #2, FWP A Feedback, and S/G 11 MFV Feedback. The rule was not applied to other analog input signals, such as

neutron flux signals, since their failures do not affect the system status, as denoted in the main CPU FMEA table of Appendix A.

With all deviation logic disabled, the number of individual failures leading to system failure increases to 179 from 112 in the base case. The total probability of system failure increases from 0.079 to 0.31 (corresponding to a frequency of 0.37 per year). The reasons for the large increase in system failure probability are due to (1) high failure rates of sensors and transmitters comparing to those of other components, (2) both validity checking and deviation checking for sensor input signals are disabled, and (3) deviation logic for both MFV and FWP demand feedback signals are disabled. If only the validity checking is disabled (i.e., the CPUs still compare redundant input signals), then the impact on system-failure probability may not be nearly as great, because some of the signal failures that were assumed to lead to system failure in this sensitivity study might instead result only in a failover to the backup CPU, as discussed in Section 2.1.

In this study, it is assumed that a drifting input signal to a CPU module will eventually fail OOR and thus can be detected by validity checking. The sensitivity calculation appears to suggest the effect of this assumption may be significant (and non-conservative). In reality, the deviation logic is always active. Therefore, even if the drifting signal does not drift OOR, the deviation checking should be able to prevent system failure (by initiating a failover to the backup CPU or switching from 3-element control to 1-element control) for all drifting signals except for those drifting level signals due to failures of level sensors or transmitters or drifting feedback demand signals due to failures of the MFV and FWP controllers (since, in these cases, both the main and backup CPUs would receive the faulty signal). However, feedback demand signals that drift OOR due to MFV or FWP failures are currently modeled as leading to system failure. Therefore, only the drifting level signals due to sensor or transmitter failure are currently treated in a non-conservative manner. As such, consistent with the last insight from the previous paragraph, the effect of this assumption is not as large as indicated by the sensitivity calculation.

The assumption on drifting signals can potentially be addressed in the future by refining the definition of drifting failure modes into two types, within and outside the range, and (1) including and accounting for the failure modes of drifting within the range in the automated tool, or (2) including plant dynamics (i.e., incorporating a model of plant response) to simulate the impact of such failure modes. Including plant dynamics could help capture the subtle timing aspects of the performance of the DFWCS. However, the drifting signal issue is likely to be difficult to address even with a model of the plant included in the automated tool.

8.4.5 Summary of Sensitivity Analyses

Table 8-2 summarizes the sensitivity analyses. It shows that the benefits of both the design of the MFV demand feedback and the external WDTs. Deviation logic offers more benefits to the system reliability than does the backup CPU because the majority of analog input failures that are identified by the deviation logic can be corrected for without failing over to the backup CPU (i.e., they would not result in system failure even if no backup CPU were present). It should be pointed that uncertainty and sensitivity analyses can be valuable for providing relative comparisons and insights when failure parameter data are of limited quality or quantity, as is the case for this proof-of-concept study. For example, in Table 8-1, the mean value is close to the point estimate, indicating that the mean estimate is not sensitive to the large uncertainties of the component failure rates used.

Table 8-2 A summary of sensitivity analyses.

Sensitivities	System failure probability of singles	System failure probability of doubles	System failure probability of triples	Total probability of all failures (initiating frequency per year)
Base case	0.051	0.023	0.0052	0.079 (0.083)
No backup CPU	0.13	0.024	0.002	0.15 (0.16)
No external WDT	0.058	0.024	0.005	0.088 (0.092)
No MFV feedback	0.051	0.023	0.0052	0.080 (0.083)
No deviation logic	0.25	0.051	0.0045	0.31 (0.37)

9. COMPARISON WITH DESIRABLE CHARACTERISTICS

Chapter 2 of NUREG/CR-6962 [Chu 2008a] presents a set of desirable characteristics for reliability models of digital systems. This section details the way the proof-of-concept model described in this report incorporates or fails to incorporate those desirable characteristics. The discussion of how the characteristics are addressed is organized in the following nine subsections corresponding to the nine categories of characteristics. First, each desirable characteristic is stated, followed by a description of how well the model meets it. The reader is encouraged to review the background information provided for each criterion in NUREG/CR-6962 [Chu 2008a], Chapter 2.

Based on the comparison of the proof-of-concept model to the set of desirable characteristics below, a number of limitations in the state-of-the-art were identified. These limitations represent areas of potential additional research as specified in Section 11.3.

9.1 Level of Detail of the Probabilistic Model

1.1 A reliability model of a digital system is developed to such a level of detail that captures the design features affecting the system's reliability, provides the output needed for risk evaluations, and for which probabilistic data are available.

As described in Chapter 3, the digital feedwater control system (DFWCS) was decomposed into three levels of detail: system, module, and component. This study defined a module as a major component that contains a microprocessor and its directly associated components. Examples of components of the modules are the analog/digital (A/D) and digital/analog (D/A) converters, microprocessors, random-access memory (RAM), read-only memory (ROM), multiplexers (MUX), demultiplexers (DEMUX), and some analog input and output devices, such as current loop devices. The failure modes for these generic components are included in the model. The level of detail of analysis in the model is characterized by the signals processed by these components, and the simulation tool that runs a slightly modified version of the software of the major modules of the DFWCS.

The level of detail of this study, i.e., at the generic component level, captures most of the design features of the DFWCS, particularly the normal behavior of the software of the system, and allows the contributions of the components to system reliability to be included in the reliability model. The model of the DFWCS allows for estimation of the frequency that loss of automatic control takes place, which is the top event and can be used in a probabilistic risk assessment (PRA), thereby satisfying the objective of demonstrating the underlying method. It is possible to envision lower levels of detail, e.g., stuck-at-one or stuck-at-zero faults, that can capture lower level design details. However, developing reliability models for the whole system at the lower levels is not likely to be feasible due to the complexity of the model and the lack of lower level failure parameters and supporting analysis tools similar to the simulation tool developed in this study.

9.2 Identification of Failure Modes of the Components of a Digital System

2.1 A method is applied for identifying failure modes of the basic components of the digital system and their impact on the system. This method provides a systematic way of carrying out this identification such that there is confidence that the failure modes obtained are as complete as possible.

Two main processes were used to identify the failure modes of the components of the DFWCS. The first process consisted of reviewing in detail the failure modes described in the plant's hazards analysis, i.e., basically the failure modes and effects analysis (FMEA) of the licensee using the system on which this case study is primarily based. The second process involved reviewing the literature about failure modes of digital components. As indicated in Section 8.3, due to limitations in the state-of-the-art, the completeness of failure modes is considered an area where more work is needed. Also, as discussed in Chapter 6, this need applies to the failure parameters needed to support reliability modeling at this level of detail. In addition, the continual advances in digital instrumentation and control (I&C) technology may significantly impact the set of component failure modes (e.g., adding and/or eliminating some failure modes) and component reliability.

The impact of each failure mode identified and of combinations of the identified failure modes was determined using the simulation tool described in Chapter 4. This tool systematically establishes the impact on the DFWCS of a very large number of these combinations. The combinations that are not studied, i.e., those having more than three failure modes, are considered to have a small or negligible contribution to the frequency of loss of automatic control of the DFWCS.

2.2 Supporting analyses are carried out to determine how specific features of a design, such as communication, voting, and synchronization, could affect system operation. These analyses determine whether the specific design features could introduce dependent failures to be modeled.

Communication between the modules of the system and between the components of each module was studied carefully, and appropriately included in the FMEA and reliability model. The use of Microlink as a means of communication has no effect on automatic control of feedwater. The DFWCS does not vote, nor carry out any significant synchronization functions. It is recognized that many digital systems, including some safety-related systems, rely on communication, voting, and synchronization to function properly, and these features have to be correctly included in their reliability models. On the other hand, the DFWCS has some important features, such as the use of watchdog timers (WDTs), and their treatment is discussed below in Section 9.4, "Modeling of Dependencies."

2.3 Failure modes that have occurred in the operating experience are examined and their applicability to the digital system being studied is considered.

A manual reactor trip occurred at the plant where the DFWCS was operating due to drifting demand signal from the main feedwater valve (MFV) controller. While the drifting signal to the main feedwater regulating valve (MFRV) is not modeled explicitly, it can be argued that this failure mode is covered by other failure modes of the MFV controller,

i.e., out-of-range (OOR) demand signals. In another incident at the plant, a maintenance-induced loss of 120v alternating current (AC) power to the main and backup central processing units (CPUs) together with an independent failure of the MFRV positioner-selector solenoid valve caused a low steam-generator level and automatic plant trip. This study models loss of power to the CPUs as a cause of loss of automatic control, which is an important contributor to the top event.

2.4 The probabilistic model of the digital system accounts for the possibility that the system may fail due to incorrect design requirements, or due to correct requirements that are not correctly implemented into the system.

This characteristic was not addressed explicitly because these kinds of failures usually are not considered in a typical reliability model, i.e., a specific failure event representing incorrect design requirements was not included in the model, though such failures could be considered to fall under the software failure placeholder events (which are discussed in Section 9.3). Further, there was insufficient information available to the study team to assess the adequacy of design requirements. On the other hand, the use of the simulation tool allows some software design issues to be identified, e.g., two potential design weaknesses were identified and discussed in Chapters 3 and 4.

9.3 Modeling of Software Failures

3.1 Software failures are accounted for in the probabilistic model.

Quantitative software reliability is beyond the scope of this study. Nevertheless, the FMEA and reliability model consider some basic software failures, such as common-cause failure (CCF) of the software of the main and backup CPUs. Two types of software failure modes are considered: software continues running but generates erroneous results, and software stops running. In addition, the simulation model accounts for the performance of software given the occurrence of one or more component failures.

It should be pointed out that a commonly accepted basis for modeling software failures probabilistically has not been established yet and additional research is needed, although it seems to be supported by previous work in Chu [2006].

3.2 Modeling of software failures is consistent with the basis of how they occur, that is, software failures happen when triggering events occur.

Qualitatively, using high-level software failure rates in the model is consistent with this basis, but considering software reliability quantitatively is beyond the scope of this study. In order to quantify the contribution of software failures, quantification methods for software reliability need to be further developed. Additional research in this area needs to be done.

3.3 Modeling of software failures accounts for the context/boundary condition in which a software is used.

This criterion was not addressed because quantifying software reliability is beyond the scope of this study.

3.4 The model of the software includes the "application software" and the "support software."

The use of high-level software failure rates does not specifically differentiate between the two types of software failures. However, in principle, the contributions from both types can be included in the failure rates.

9.4 Modeling of Dependencies

Dependencies Due to Communication

4.1.1 Inter-system failure propagation is addressed, and modeled as applicable.

The plant has a separate DFWCS for each of its two loops, but the scope of this proof-of-concept study covers only one DFWCS. The interfaces of the DFWCS with other systems through the system's input and output are modeled by considering failure modes of the associated signals. For example, the model covers the two DFWCSs' exchange of MFV demand signals that are used in calculating feedwater pump (FWP) demand. However, the model excludes the reactor trip and turbine trip signals received by the system; their failures are omitted because they would be addressed as separate initiating events in a plant PRA.

4.1.2 Inter-channel failure propagation is addressed, and modeled as applicable.

The DFWCS does not have "channels." However, it has two redundant CPU modules that can be interpreted as "channels." Hence, the interactions between the main and backup CPUs were studied in detail, and the potential propagation of failures between them was considered in the FMEA and reliability model. The simulation tool accounts for propagation of failures within the DFWCS.

4.1.3 Intra-channel failure propagation is addressed, and modeled as applicable.

The DFWCS does not have redundancy within each of its modules; hence, this criterion is not applicable.

Dependencies Due to Support Systems

4.2.1 Loss of power to safety-related digital systems is modeled. It is important to note that there may be cases where loss of power generates an actuation signal, i.e., the system or component fails safe. If this is the case, loss of electric power is not modeled as a cause of failure on demand of the system or component. Instead, it is modeled for the generation of a spurious signal.

The dependencies of the modules of the system on electrical power were considered and included in the FMEA and reliability model.

4.2.2 If dependencies on heating, ventilation, and air conditioning (HVAC) are relevant, they are modeled.

The dependencies on HVAC were not considered significant, as stated in Section 2.6.

4.2.3 Other potential dependencies on support systems are considered, and modeled as applicable.

No other relevant dependencies were identified.

<u>*Dependencies Due to Sharing of Hardware*</u>

4.3.1 The digital systems of a plant are examined to determine if there are dependencies due to sharing digital hardware. Any relevant dependencies are modeled.

Dependencies due to sharing hardware were identified and covered in the FMEA and reliability model. An example of this type of dependency is that during normal operation, the MFV, the bypass feedwater valve (BFV), and the FWP controllers use the demand signals from the main CPU. The failure of this CPU would affect these three controllers. If the failure of the main CPU is detected, a failover to the backup CPU would occur; then, these three controllers would depend on the signals from the backup CPU.

4.3.2 The effect of sensor failures on the digital system and on other components or systems of the plant are evaluated and included in the probabilistic model.

The main and backup CPUs receive signals from plant sensors that are common to both CPUs. The FMEA and reliability model include this dependency on sensors.

4.3.3 The failures of devices that process the output of redundant channels of a system are modeled.

The DFWCS does not have “channels,” but has two redundant CPU modules that can be interpreted as “channels.” The controllers of the DFWCS, such as the MFV and FWP, process the output of the two. The FMEA and reliability model include failures associated with the controllers.

4.3.4 Failure of a digital system may trigger an initiating event with possible additional failures of mitigation features. This dependency also is included in the model, as applicable.

The reliability model of the DFWCS is intended to evaluate the frequency of the initiating event, “loss of automatic control by the DFWCS.” The scope of this proof-of-concept study did not include considering a degradation or loss of features that offer mitigating capabilities after an initiating event; hence, this criterion is not applicable.

Modeling of Fault-Tolerant Features

4.4.1 The deterministic analysis of the digital system identifies those failure modes of a component that the fault-tolerant features can detect and the system is able to reconfigure itself to cope with the failure. The probabilistic model only credits the ability of these features to automatically cope with these specific failure modes. It considers that all the remaining failure modes cannot be automatically tolerated.

Each failure mode of the components of the modules was explored fully to identify those failure modes that the fault-tolerant features of the DFWCS can detect, e.g., WDTs, OOR and rate of change checks of analog signals, feedback of the controller output signals, and exchange of status information among the modules. The probabilistic model only credited the ability of these features to cope automatically with the appropriate specific failure modes. In the model, the remaining failure modes cannot be automatically tolerated.

Note, while the fault-tolerance features of the DFWCS are accounted for in this study, other such features (such as different hardware redundancy techniques and software fault-tolerance design) can be applied to digital system designs. The ability to account for other fault-tolerant features remains to be demonstrated.

4.4.2 When applying a value of "fault coverage" to the probabilistic data of a component, the types of failures that were employed in the testing used to derive this value are known. No credit for fault coverage is given to those failure modes that were not included in the testing. This also would apply when using a value of fault coverage from a generic database or the literature.

As mentioned above, whether or not a component failure mode can be detected was assessed based on the available design information. The probabilistic model did not use values of "fault coverage" and so this criterion is not applicable. In this study, for each failure mode associated with a CPU module which has an independent WDT, plant information and an understanding about how the system works were used to determine if the effect of each failure mode on the module can be detected by its WDT and/or the application software. Some failure modes are considered detectable and others are not. The probability that an individual failure mode or sequence is detected by the WDT was assumed to be either one or zero given that the WDT functions properly. In this sense, the coverage is automatically accounted for in the probabilities of all failure sequences. However, due to limitations in the state of the art for FMEA, whether or not the failure modes of some components, such as a RAM, can be detected by the fault tolerance features was determined subjectively. The concept of fault coverage can be used to improve this treatment. In general, fault coverage can be used to adjust the component failure rates, as in Aldemir [2009].

4.4.3 Information from a generic database about a specific probabilistic datum of a component, such as a failure rate, is reviewed to assess whether it was adjusted for the contribution of fault coverage. If so, this datum may be used in a probabilistic model, but no additional fault coverage is applied to this component, unless it can be shown that the two fault coverages are independent.

As stated above, this criterion is not applicable, as the probabilistic model did not use "fault coverage."

4.4.4 A fault-tolerant feature of a digital system (or one of its components) is explicitly included either in the logic model or in the probabilistic data of the relevant components, but not in both.

The system logic model covers fault-tolerant features of the DFWCS, i.e., these features are accounted for by the combinations of failure modes that fail the system identified via the process described in Chapters 3 and 4. Because of the lack of information about the data of PRISM [Reliability Analysis Center (RAC) Manual], it cannot be determined if any of the modeled fault-tolerant features are built into the data.

4.4.5 The probabilistic model accounts for the possibility that a fault-tolerant feature may fail to detect and/or fix a failure mode that it was designed to catch.

The software of the main and backup CPUs implements some fault-tolerant features. To some extent, using a simulation tool automatically captures software faults. In addition, the model includes software failure rates as placeholders. Since the scope of this study does not cover quantitative software reliability, software failures associated with fault-tolerant features were not explicitly considered. On the other hand, the fault-tolerant features implemented in hardware were studied, e.g., WDTs, and their failures were considered in the FMEA and reliability model.

4.4.6 If the detection of a failure of a component depends on other components, e.g., a WDT, then the dependency is modeled.

The detection of some failure modes of some components of the DFWCS depends on other components, such as a WDT. This dependency was included in the FMEA and reliability model.

4.4.7 The probabilistic model accounts for the possibility that after a fault-tolerant feature detects a failure, the system may fail to re-configure properly, or may be set up into a configuration that is less reliable than the original one.

The FMEA and reliability model account for the possibility that after a fault-tolerant feature detects a failure, the configuration of the DFWCS changes into one that is less reliable than the original. For example, if a failure mode of the main CPU is detected, a failover to the backup CPU will occur, and the DFWCS will lose the original redundancy afforded by the main and backup CPUs. Judgment/understanding of the system was used to determine whether a WDT could detect a failure mode. In many cases, the failure modes are defined such that their detectability is simple to determine, e.g., a failure mode for sensor, OOR high signal can be detected by an OOR check of the CPUs.

Dependencies Related to Type I and II Interactions

4.5 The probabilistic model addresses Type I and Type II interactions.

This study did not specifically address Type I interactions (interactions with controlled processes), but considered Type II interactions (interactions among the components of the digital system) by studying the failure modes related to some events, such as communication between different components and multiplexing. The inability to model the Type I interactions was discussed in Sections 3.4 and 4.4 in detail, and relates primarily to modeling of drifting signals. This limitation does not appear to have a significant impact on the results.

Dependencies Related to CCFs

4.6.1 Intra-system hardware CCF. Hardware CCF between similar components within a system is modeled.

Hardware CCF between similar components of the DFWCS was considered; for example, that between the main and backup CPUs. Due to lack of digital-specific CCF data, a generic beta factor was used. Collection of CCF data on digital components is an area where additional research is needed.

4.6.2 Intra-system software CCF. If the channels or subsystems of a digital system (and/or the redundancy within a channel or subsystem) use similar software, software CCF is modeled.

As mentioned above, software failures, including software CCF, are beyond the scope of this study. Nevertheless, the FMEA and the reliability model consider some software CCFs, such as the CCF of the software of the main and backup CPUs. Many risk analysts believe that software CCFs are the most risk significant failures for digital I&C systems. This is an area for additional research.

4.6.3 Inter-system hardware CCF. Hardware CCF between different systems using the same hardware is modeled.

This proof-of-concept study is limited to a single DFWCS; hence, this criterion is not applicable.

4.6.4 Inter-system hardware CCF. If similar software is used in different digital systems, software CCF is modeled.

This proof-of-concept study is limited to a single DFWCS; hence, this criterion is not applicable.

Note, however, that inter-system software CCFs may occur for a digital system, e.g., the same support software (operating system, platform software, etc.) may be used across the system boundaries. Modeling of inter-system software CCF is an area where additional research is needed.

9.5 Probabilistic Data

Probabilistic Data for Hardware

As discussed in Chapter 6, publicly available hardware failure databases of digital components are limited and have very large uncertainties. Hardware failure data is an area where additional research is needed. The following describes how this study addresses the desirable characteristics.

If component-specific data are available, they should satisfy the following criteria:

5.1.1 The data are obtained from the operating experience of the same component as that being evaluated, and preferably in the same or similar application and operating environment.

Component-specific data were unavailable for this study; hence, this criterion is not applicable.

5.1.2 The sources of raw data are provided.

As stated above, component-specific data were unavailable; hence, this criterion is not applicable.

5.1.3 The method used in estimating the parameters is documented, so that the results can be reproduced.

As stated above, component-specific data were unavailable; hence, this criterion is not applicable.

If component-specific data are not available, generic data, i.e., from a generic database, may be used as long as they satisfy the following criteria:

5.1.4 The data of the same generic type of component are used and wide uncertainty bounds are expected.

This study used the raw data of some digital components from the RACdata database of PRISM [RAC manual] in a Hierarchical Bayesian analysis to account for the variability of data sources; very large error factors were obtained for some of the failure parameters. Point estimates of the RACRate model of PRISM with large assumed error factors were used for other components. However, the dearth of information on the definition of components in the raw data, and on its sources, raises issues about the applicability of the data to the components in this study. Failure-mode distributions are another type of failure data that are needed in this study. One problem associated with the failure-mode distributions used in this study is that they do not include all of the applicable failure modes for some components. Given these limitations in the data, its use in this study was intended only to demonstrate the proposed approach and exercise the model.

5.1.5 It is verified that the generic data were collected from components that were designed for applications similar to those in nuclear power plants (NPPs).

As discussed in 5.1.4, the applicability of PRISM's raw data and failure rate estimates is a concern.

5.1.6 The sources of the generic database are given.

PRISM is the source. As discussed in 5.1.4, there is little information about the sources of raw data included in the RACdata database of PRISM.

Both component-specific and generic data should meet the following criteria:

5.1.7 If the system being modeled is subject to an adverse environment and the data are obtained from systems that are not subject to a similarly adverse environment, then the data is modified to account for the corresponding impact of the specific environment on the reliability of the system components.

The components of the DFWCS are located in environments that are not normally adverse, i.e., the NPP control room and auxiliary building.

5.1.8 Data for CCFs also address the above characteristics*.*

There are no CCF parameters for the DFWCS components. A generic beta factor was used from the Advanced Light Water Reactor Utility Requirement Document [EPRI 1993].

5.1.9 Data *for "fault coverage" also address the above* characteristics*.*

As mentioned above, whether or not a component failure mode can be detected was assessed based on available design information. The probabilistic model did not use values of "fault coverage" and so this criterion is not applicable.

Note, due to limitations in the state of the art for FMEA, whether or not the failure modes of some components, such as a RAM, can be detected by the fault-tolerance features was determined subjectively. The concept of fault coverage can be used to improve this treatment.

5.1.10 Documentation of basic event calculations includes how the basic event probabilities are calculated in terms of failure rates, mission times, and test and maintenance frequencies.

The component failure modes are represented by failure rates and used in calculating failure-sequence probabilities and frequencies. The latter are obtained using the solution of the Markov model. For a single-failure sequence, failure probability is calculated as the probability that the failure mode occurs (and no other failures occur) in one year. For double-failure sequences, the failure probability is the probability that one failure occurs followed by the second failure during one year. Triple-failure sequences are quantified similarly.

Probabilistic Data for Software

5.2 A method for incorporating the contribution of software failures to digital system unreliability is used and documented.

As mentioned above, this criterion is not applicable as quantification of software reliability is beyond the scope of this study (since it is not within the current state of the art). Arbitrary failure rates were assumed for the placeholder software failure events included in the model.

9.6 Uncertainty

6.1 Uncertainties associated with the probabilistic data for hardware and software are estimated.

Uncertainties associated with failure parameters of the model were estimated and used in an uncertainty analysis of the top event.

6.2 Parameter uncertainty is propagated throughout the PRA model such that the uncertainty characteristics of the risk measures, such as core damage frequency, can be determined.

Uncertainties associated with failure parameters of the model were propagated to obtain an estimate of the uncertainty of the top event.

6.3 Key assumptions of the model are identified, and a discussion of the associated model uncertainty provided, including the effects of alternative assumptions.

A few assumptions were identified with the associated modeling uncertainty discussed, along with alternative assumptions, e.g., plant dynamics and modeling drifting signals. They are documented in Section 8.2 and elsewhere throughout the report.

9.7 Integration of the Digital System Model with a PRA Model

7.1 For full effectiveness of the digital system reliability model, it is possible to integrate it into the plant PRA model; the process for integration is verifiable.

Integrating the reliability model of the DFWCS with a PRA model is beyond the scope of this study; hence, this criterion is not applicable.

7.2 If a model of a digital system has been integrated with a PRA model, all the dependencies related to the system are accounted for. They are the dependencies of the digital system on other systems (such as its support systems), and of other systems on the digital system.

As mentioned above, integrating the reliability model of the DFWCS with a PRA model is beyond the scope of this study; hence, this criterion is not applicable.

9.8 Human Errors

8.1 Human errors during upgrade of hardware and software are included.

The analysis of human errors associated with the DFWCS is beyond the scope of this study; hence, this criterion is not applicable.

8.2 Human errors related to human system interface are included.

The analysis of human errors associated with the DFWCS is beyond the scope of this study; hence, this criterion is not applicable.

In this study, a loss of automatic control of the DFWCS is defined as a system failure. It should be recognized that operator action may still be able to save the system from a loss of automatic control and maintain the feedwater level manually without causing an initiating event. In addition, different failure modes may generate different alarms and/or annunciations, which are likely to affect performance of the operator in a different way. Additional research in this area will help create a more realistic reliability model.

9.9 Documentation and Results

9.1 Key assumptions made in developing the reliability model and probabilistic data are documented.

The key assumptions made in developing the reliability model were documented throughout this report, particularly in Chapters 2 through 7.

9.2 Assumptions made in developing the reliability model and probabilistic data are realistic, and the associated technical justifications are sound and documented.

Most of the assumptions made in developing the reliability model are realistic, and the associated technical justifications are robust; both were documented throughout this report. In a limited number of cases, arbitrary assumptions were made due to a lack of information or data; these assumptions are typically not expected to significantly affect the results of the analysis. For example, in the case of assuming “isolated” failure of input and output signals, the issue is not judged of much concern due to a general lack of redundancy in the system. It may be more important for other systems. On the other hand, in the current study, both the BFV and pressure differential indicating (PDI) controllers are not modeled, which may affect the overall system failure probability. As discussed in Section 4.4, if the CPU digital input that represents the BFV automatic/manual status indicates “manual,” this will cause a system failure and is accounted for in the study. However, a number of component failures in the BFV controller that may cause the same impact as that failure are not included in this study. Similarly, a spurious takeover of the MFV controller by the PDI controller also causes a system failure. Analyzing the BFV and PDI controllers to identify potential causes of the spurious failure of the BFV automatic/manual status and the spurious takeover by the PDI controller would be needed to better estimate the overall system failure probability.

9.3 The dominant failure modes *of the reliability model are documented with a description of the sequence of events that need to take place and how the failures propagate to fail the system. The sequence of events realistically represents the systems behavior at the level of detail of the model.*

The dominant sequences identified are consistent with the system behavior at the level of detail of the model. However, understanding that the failure parameters used in the quantification are weak, data on dominant sequences are not included in this report.

10. COMPARISON OF RESULTS WITH THOSE FROM DYNAMIC METHODS

This chapter documents a comparison of the results and insights obtained in this study using "traditional" methods for the failure of the digital feedwater control system (DFWCS) with the results from the application of two "dynamic" methods described in NUREG/CR-6985 [Aldemir 2009]. As stated previously, dynamic methods are defined as those that attempt to explicitly model the interactions between a digital instrumentation and control (I&C) system and the plant physical processes.[9] The comparison is performed at a high level and covers the scope and level of detail of the models and their qualitative results and insights. Due to differences in top event definition and boundary conditions and, as stated in Chapter 7, weakness in failure parameters, no comparison is made between the quantitative results of the different studies. The dynamic methods employed were the Markov/cell-to-cell-mapping technique (CCMT) and the dynamic flowgraph methodology (DFM). Section 10.1 briefly describes the application of these dynamic methods in evaluating the DFWCS reliability. Section 10.2 is a short comparison of the DFWCS models incorporating dynamic methods and the traditional method of this study. Finally, Section 10.3 explores the qualitative results obtained from the different models.

10.1 Application Of Dynamic Methods to the DFWCS

Chapter 1 of NUREG/CR-6985 [Aldemir 2009] summarizes the scenario for modeling the DFWCS as a plant transient, and its failure as "…either a low or high steam generator level event, normally followed in both cases by a turbine and reactor trip." Accordingly, the failure of the DFWCS is defined as having two possible undesirable outcomes (named "top events"): low level in the steam generator (S/G) and high level in the S/G. The power transient is assumed to be initiated by an operator manually controlling reactor power using the control rods. The plant transient is produced by a power maneuver consisting of:

1. power ramp-up, starting from 70% of full power,
2. steady-state at 78% of full power, and
3. power ramp-down, back to 70% of full power.

The maneuver constitutes good application ground because it exerts and challenges the main function of the DFWCS, i.e., maintaining the S/G water level between set limits under changing power demand. A 24-hour period was chosen because it is the default reference-time period for standard probabilistic risk assessment (PRA) tools when modeling continuously operating systems. This time period was equally divided among the three phases.

Chapter 4 of NUREG/CR-6985 [Aldemir 2009] describes an attempt to consider the contribution of the failure of a DFWCS to the failure to mitigate an initiating event (IE), i.e., turbine trip. In other words, an existing event tree for the IE turbine trip of a two-loop pressurized water

[9] As discussed in Chapters 3 and 4, in order to identify all of the DFWCS component-level failure mode sequences, it was necessary to augment the traditional methods through the use of a simulation tool. Nonetheless, in this report, the methods applied are still referred to as "traditional," since they do not attempt to explicitly model the interactions between the DFWCS and the plant physical processes.

reactor (PWR) was modified to cover the failure of main feedwater system (MFW) caused by failures of the DFWCS. As mentioned above, the first eight hours of the plant transient consist of a linear power ramp-up from 70% to 78% of full power. The results from the two dynamic methods during this ramp-up are used for comparing with the results obtained using the traditional method.[10]

10.2 Comparison of Scope and Level of Detail

This section offers a high-level comparison of the scope and level of detail of the two studies; it is not intended to be a thorough detailed comparison of their models.

1. The most important difference between the two is that the dynamic methods, by definition, consider the interactions of the DFWCS with the plant processes by including simplified thermal hydraulic models of these processes. In contrast, the traditional method only implicitly considers the plant condition, i.e., full power operation. In other words, the plant condition determines the sensor inputs to the simulation tool for the DFWCS. The inability to model the Type I interactions was discussed in Sections 3.4 and 4.4, and primarily relates to modeling of drifting signals. This limitation does not appear to have a significant impact on the results.

2. The dynamic methods define system failure in terms of level in the S/G, while this traditional method study defines system failure as loss of automatic control, which can be determined in terms of the internal properties of the DFWCS, e.g., the main feedwater valve (MFV) controller enters “manual” mode.

3. Dynamic methods develop models of the software of the DFWCS that represent the normal behavior of the software, and try to capture potential software faults by developing different scenarios/boundary conditions that may challenge it. The models of the software developed using dynamic methods approximately model the control law and the complex logic that is used in the software to process digital signals, such as status information. The traditional method study uses the simulation tool that runs the actual software. The simulation tool contains complex status logic, in addition to control laws, to account for the system response to many postulated hardware failures and their combinations. The simulation runs also may capture potential software faults as exemplified in the two types of scenarios described in Chapters 3 and 4 (i.e., the scenario involving MFV controller output fails low and the scenario involving MFV controller sending an incorrect main central processing unit (CPU) status to the main CPU).

4. The dynamic methods employ “coverages” estimated by a fault injection method based on an emulator of the main and backup CPUs. These coverages are measures of fault-tolerance features internal to the microprocessors of the CPUs, and are applied as reduction factors for the failure rates used in the dynamic reliability models. The fault injection method captures a very low level of design detail and allows the development of a much higher-level reliability model. Its limitation is that it only measures the

[10] The comparison of the dynamic methods’ results to the traditional method results was performed prior to the completion of NUREG/CR-6985 [Aldemir 2009]. Following this comparison, additional results were obtained from application of the dynamic methods that cover the ramp down period. These additional results were not compared to the traditional method results.

microprocessors' response to injected faults. It does not consider interactions between the CPUs and other system components, e.g., demand feedback from the controllers and subsequent failover. The simulation tool used to augment the traditional method models the whole DFWCS, and can consider failure modes associated with any components that are part of the DFWCS, e.g., the sensors, analogs/digitals, multiplexers and watchdog timers (WDTs). It also can account for, in an integrated way, failure modes at a lower level than those of the dynamic models, e.g., a spurious signal from the WDT indicating that the CPU has halted.

5. The dynamic models include the failure modes of the "actuated devices," e.g., the main feedwater-regulating valve (MFRV) "stuck" in its current position. In contrast, the traditional methods study modeled the valve positioners and pump speed controllers that are digital components in a simplified way, i.e., as single components, since it was considered that the non-digital components of the system can be easily modeled as part of a conventional PRA.

6. The dynamic models are more subject to the potential state explosion issue if too many components and processes are introduced into them. The traditional approach Markov model is developed and quantified by treating individual sequences separately, i.e., each sequence is simulated and quantified separate from those of other sequences; in this way, linear scalability is achieved, i.e., the sequences can be simulated and quantified by running multiple computers in parallel, and combining the results later.

10.3 Comparison of Results from Traditional and Dynamic Methods

As described in detail in Chapter 3 of NUREG/CR-6962 [Chu 2008a], degradation or total loss of the MFW system has two contributions to plant risk: (1) It may cause IEs and (2) it may fail to fulfill its mitigative function after a reactor trip. As discussed in NUREG/CR-6962 [Chu 2008a], the first contribution is analyzed in this study because it is considered more significant to plant risk. Hence, the traditional method was used to develop a model for failures of a DFWCS that cause an IE.

As described in Section 10.1, the dynamic methods modeled a DFWCS during a plant transient between 70% and 78% of full power, but proposed to use it to model the mitigation of an IE, i.e., with the plant shut down, and a DFWCS operating in low-power mode. Nonetheless, the ramp-up (first eight hours of the plant transient described above) proposed by NUREG/CR-6985 [Aldemir 2009] is similar to the condition of the plant assumed by the traditional method, i.e., full power. Accordingly, the qualitative results from the traditional method can be roughly compared with those from the dynamic methods, recognizing the differences in boundary conditions between the respective models.

In this report, the failure of a DFWCS is defined as loss of automatic control of feedwater within one year (given that the plant is at full power), while the definition of failure of the DFWCS (during the 8-hour ramp-up) in the dynamic method models is expressed as two top events: "low level in S/G" and "high level in S/G." Since the occurrence of each top event constitutes a loss of automatic control, the results for each can be qualitatively compared to those of the traditional method. The main qualitative results from the three methods (i.e., Markov/CCMT, DFM, and traditional) are the combinations of failure modes that cause loss of automatic control.

A one-to-one comparison of the combinations of failure modes that cause system failure obtained by each method is beyond the scope of this task. Nevertheless, the following are the main insights gained from comparing the results of both dynamic methods with those of the traditional method:

1. Due to the detailed, comprehensive approach used for implementing the traditional method, the resulting combinations of failure modes that cause system failure appear to include the combinations identified by the dynamic models. However, NUREG/CR-6985 [Aldemir 2009] only provides the most dominant (i.e., most likely) failure modes combinations. Also, it is possible that a different application of the dynamic methods may produce some combinations of failure modes that the traditional method may not be capable of identifying.

2. The scope of the dynamic models included the failure modes of "actuated devices," e.g., the MFRV "stuck" in its current position, while the traditional model did not include them. Accordingly, the former models encompassed some combinations (of failure modes causing system failure) involving the failure modes of these devices that the traditional model did not identify. However, the traditional method can obtain these combinations by modeling the non-digital components of the system as part of a conventional PRA.

3. The traditional method, as applied in this study, identified system failure resulting from combinations of failure modes of detailed components of the DFWCS; the dynamic models used failures modes at a coarser level, i.e., similar to the module level as defined in this study.

4. The dynamic methods assumed that the two CPUs (main and backup) and the three controllers (MFV, bypass feedwater valve, and feedwater pump) share the same power source. Accordingly, its loss is a major contributor to their results. However, this assumption seems to be a conservative one. In this study, a more realistic power supply arrangement was assumed by modeling separate power sources for the CPUs and controllers.

5. Quantitatively, this study obtained a probability of approximately 0.1 for system failure in one year of operation that converts to approximately $1x10^{-04}$ for an 8-hour period. This value is of the same order-of-magnitude as the results of the dynamic methods. It is difficult to compare the detailed results, partly because the sequences obtained using the traditional method tend to be at a lower level of detail, and partly because different failure parameters seem to have been used.

11. CONCLUSIONS, INSIGHTS, AND AREAS OF POTENTIAL ADDITIONAL RESEARCH

This study develops an approach for modeling digital systems using the Markov method and applies it to a digital feedwater control system (DFWCS) to demonstrate the underlying concepts of the approach. The top event is the loss of automatic feedwater control. A failure modes and effects analysis (FMEA) was performed at a relatively fine level of detail, e.g., at the level of multiplexers (MUXs) and analog/digital (A/D) converters. This level of detail is considered appropriate for supporting the proof-of-concept reliability analysis of the DFWCS. The FMEA approach used in this study should be applicable to other digital systems, though the level of detail of the FMEA will be a function of the particular study objectives. The study uses publicly available data on the failure modes of the hardware components, thereby allowing important design features to be properly accounted for. Model development includes development of a simulation tool that simulates the execution of the DFWCS software. The simulation tool is used to determine the system response to postulated hardware failure modes and combinations thereof. The sequences of component failure modes that lead to a system failure are then used in defining the sequences of transitions in a Markov model. The Markov model is quantified to estimate the annual frequency with which a loss of automatic control of feedwater takes place, and to support sensitivity calculations that evaluate the benefits and importance of some of the features of the digital design, such as watchdog timers (WDTs), feedback of demand signals, and deviation logic. The quantification of the system model makes use of publicly available component failure parameters and the results of a Hierarchical Bayesian Method (HBM) analysis [Yue 2006] of the raw data in the PRISM database [Reliability Analysis Center (RAC) Manual] that accounts for the uncertainty associated with different data sources.

The following is an outline of the procedure of the demonstrated approach:

1. Define system boundary and top event.

2. Decompose system into modules and components to a level of detail where failure data are available.

3. Perform component-level FMEA of individual failures manually to determine their effects on the system in terms of the output signals of the components.

4. Develop and validate an automated FMEA tool by (1) simulating the impacts of individual failures and comparing to the manual FMEA results and (2) simulating higher order sequences and spot checking the results, if needed, as determined by Step 5.

5. Quantify the sequences generated in Step 4, starting with individual failures, to determine if system failure probability has converged. If not, continue Step 4 with higher order sequences.

6. Proceed with subsequent steps, such as uncertainty and sensitivity analyses, as needed.

It should be emphasized that since the objective of this study was only to identify the existing capabilities and limitations of using traditional probabilistic risk assessment (PRA) methods for

developing and quantifying reliability models of digital systems, the study did not generally involve advancements in the state of the art (with the possible exception of the use of the simulation tool). Therefore, there are a number of key areas that need to be addressed (as identified in Section 11.3) before the methods described in this report can be used to support decision-making (e.g., regulatory decisions or design changes). In particular, many risk analysts believe that software common-cause failures (CCFs) are the most risk significant failures for digital instrumentation and control (I&C) systems. Due to limitations in the current state of the art, software CCFs are beyond the scope of this study.

It should also be pointed out that even though this study models a control system, the approach of this study may be applicable to protections systems, such as a reactor protection system (RPS). The conclusions and insights of this chapter are mostly related to modeling methods, and are applicable to both control and protection systems, unless otherwise specified.

11.1 Conclusions

The following conclusions are derived from performance of this study.

1. *The traditional method used in the study, i.e., Markov method, must be supported by strong engineering knowledge and supporting analyses of the systems being studied. A simulation model of the system is a critical tool in facilitating reliability model development.*

At the level of detail considered, the study requires a deterministic model that simulates the execution of the system software to capture the system design features, particularly those of the software, and to determine which sequences of postulated component failure modes would cause the system to fail. The simulation model allows the system behavior under failure conditions to be approximately accounted for in the reliability model, including not only the system control algorithms, but also the complex control logic based on the status of various signals of the controlled processes and that of the components of the system.

The important role of the simulation tool in determining system success or failure reduces the Markov methods to methods solely for quantifying system reliability (i.e., the Markov methods are not used to identify the system failure paths, they are only used to quantify them). Without the simulation tool, in practice, it would be very difficult, or even impossible, to directly develop a Markov model that captures all of the details of the system design. Although an automated tool is used, the methods applied are still referred to as “traditional,” since they do not attempt to explicitly model the interactions between the DFWCS and the plant physical processes. The Markov model formulated for this study (using the output of the simulation tool) does represent a good model of the system failure behavior, i.e., it explicitly models the order in which failures occur, and supports the derivation of simple analytical solutions.

2. *The level of detail of the DFWCS model is adequate for capturing many of the system design features, while not being too complicated to be developed and solved.*

The Markov model of the DFWCS demonstrated the feasibility of the proposed approach. Although the intent of this study is to use state of the art traditional methods to develop a reliability model of the system, the need to model realistically the DFWCS features necessitated developing a simulation tool, an enhancement to the state of the art. As discussed previously,

using a simulation model supports the development of a more realistic reliability model. Also, the level of detail of the model is consistent with that at which failure parameters are available (although the data has weaknesses, as discussed next). Even though the simulation tool does not encompass a thermal-hydraulic model of the plant, the system failure modes and sequences can be identified from information on its design. The state explosion problem of a detailed Markov model is resolved by truncating the higher order failure sequences when convergence is achieved. This process is similar to that used in a traditional PRA where the quantification process is truncated based on a cutset size limit and/or probability/frequency limit. The usefulness of the DFWCS model developed for this study is demonstrated further by performing a few sensitivity calculations that evaluate the importance of some of the digital design features, e.g., WDTs.

3. Failure parameters of digital components are scarce, and additional data are needed.

NUREG/CR-6962 [Chu 2008a] includes a review of publicly available failure data on digital systems and a Bayesian analysis of raw data extracted from the PRISM database to account for the variability in the sources of the data. That review identified and discussed some weaknesses and limitations of the publicly available databases, though no attempt was made to validate or invalidate them. The limitations of these failure parameters of digital components point to the need for additional research and development in this area. The Bayesian analysis resulted in some failure parameters with very large error factors, demonstrating large variability in the data. It may be challenging to calculate meaningful failure rate for hardware components because of this large variability. The information documented in Chapter 6 of this report is extracted from NUREG/CR-6962 [Chu 2008a], and is used only to demonstrate the reliability method and exercise the reliability model. These data are not appropriate for quantifying models intended for use in supporting decision-making (e.g., regulatory decisions or design changes). In general, data should be collected from the manufacturers of the components being modeled or from the same type of components in a similar application. It should be noted, however, that the manufacturers of components often change throughout the lifecycle of the product and component failure data is often not available or difficult to obtain. It is possible to address these limitations in the uncertainty treatment of the data analysis, though the resulting uncertainty may be very large.

11.2 Insights

A number of insights were obtained through performance of the DFWCS benchmark study. These are summarized below.

- This study found that, for the DFWCS, the order in which component failure modes occur can affect the impact the failures have on the system. For example, an individual failure that fails the system may not do so if it occurs subsequent to another failure. This is believed to be a generic feature of digital systems, and should be captured in reliability models. The Markov method can easily account for the order in which component failure modes occur by considering different orders in different sequences.

  In addition, in the above example, if the sequence of two failures does cause the system to fail, then the double sequence needs to be included as a valid sequence even though a single failure would have caused system failure because the definition of the single failure precludes the double sequence. In other words, the probability of the single failure is the probability that it occurs in one year and no other failures occur.

Accordingly, the concept that non-minimal cutsets do not need to be considered is not applicable to the sequences that cause a system failure. For this and other reasons, use of Markov quantification methods raises some issues with regard to integration with a PRA that is based on the event tree/fault tree (ET/FT) method. A number of studies have discussed how prime implicants and/or non-minimal sequences can be integrated into ET/FT models [Aldemir 2007, Aldemir 2009]. In addition, some PRA software is being modified to better address this issue. An assessment of the challenges and potential solutions for integrating the DFWCS model with a PRA is beyond the scope of this study, but will need to be addressed in the future.

- Performing the FMEA and running the simulation tool revealed two kinds of scenarios (one involving differences in signal delay times and the other involving both central processing units (CPUs) operating in tracking mode) that represent potential weaknesses of the system design. These scenarios are described in Sections 3.3.4 and 4.3. The discovery of these scenarios, which were not identified in the plant's hazards analysis, suggests that the simulation tool potentially could serve to verify and validate the system software. Including a thermal-hydraulic model of the plant would make it a more complete tool. Development of the simulation tool offers a capability to undertake test runs of the software and support deterministic evaluations of digital systems.

- The model developed for the DFWCS is significantly more detailed than that of many other studies of digital systems, e.g., those models proposed by Rouvroye [1999]. The experience of this study shows that it is difficult to capture the detailed interactions among the components and combinations of failures of the components using higher level modeling. It may be possible to use the detailed model of this study to develop an equivalent or approximate module level model by grouping the component failure modes of a module based on their impacts, e.g., on the input and output signals of the modules. Failure modes of the modules could then be defined in terms of the component failure mode groups, and used in developing a system level model in the form of high-level Markov models or fault trees.

- In developing an automated FMEA tool, it is desirable to use the source code which should be available to the nuclear power plant but may not be available to the United States Nuclear Regulatory Commission or its contractors. If the source code is not available, an FMEA tool can still be developed using design information, such as a functional description of the software, although the tool will not be as realistic as a tool developed using the source code and may not be suitable for use in studying a system in detail. In either case, if the tool is to be used for a regulatory application, it would need to be subjected to systematic verification and validation.

- The FMEA tool may have difficulty in accurately addressing the timing issue associated with the time when an additional failure occurs given one or more failures have taken place and the system has not failed yet. The FMEA tool assumes that the system is in a steady state before any failure occurs. If an additional failure occurs after the control system and the controlled processes have again reached a steady state condition after the transient caused by preceding failure or failures, then the automated FMEA tool can correctly determine the system response. If the additional failure occurs before the system reaches a steady state subsequent to the preceding failure(s), the impact of the additional failure on the system cannot be captured by the FMEA tool, because the

FMEA tool does not have a model of the controlled process and is not able to determine the transient response. It is expected that the duration of the transient subsequent to the postulated failure or failures is very short compared with the duration of one year, and the occurrence of the additional failure during the transient is very unlikely, given the assumption that the failures are independent of each other. Therefore, ignoring the transient period should not have a significant impact on the results.

- This study did not specifically address Type I interactions (interactions with controlled processes external to the digital system), but considered Type II interactions (interactions among the components of the digital system) by studying the failure modes related to some events, such as communication between different components and multiplexing. The inability to model the Type I interactions was discussed in Sections 3.4 and 4.4 in detail, and relates primarily to timing issues and modeling of drifting signals. This limitation does not appear to have a significant impact on the results.

  In this study, the simulation tool cannot properly calculate analog signals due to the lack of a thermal-hydraulic model of the plant to provide feedback signals. On the other hand, the application of two "dynamic" methods to the DFWCS system described in NUREG/CR-6985 [Aldemir 2009] explicitly models the controlled process to determine the dynamic behavior of the feedwater system. Including plant dynamics could help capture subtle timing aspects of the performance of the DFWCS, e.g., issues associated with timing of failure sequences and the impacts of a within-the-range drifting signal. However, these issues are likely to be difficult to address even with a model of the plant included in the automated tool. For example, in the case of a drifting signal, the failure impacts are affected not only by how the signal drifts, but also by the system operating point when the failure occurs. A subtle deviation in the drifting signal may cause completely different responses. In addition, it is not clear, at present, whether the increased accuracy of modeling obtained through incorporation of a plant dynamics model would justify the increased complexity. Obtaining the needed failure rate data may also be difficult.

  Unfortunately, due to differences in top event definition and boundary conditions, comparison of the results of the DFWCS studies using traditional and dynamic methods does not provide insight into the importance or benefit of incorporating a plant dynamics model in determining the DFWCS failure rate or probability.

- The proposed approach of this study may also be capable of modeling safety related protection systems, such as a RPS. For protection systems, it is believed that the use of dynamic methods may not offer any considerable improvements, because once a protection system is actuated, the feedback from the plant has no effect on the actuation. An RPS has higher redundancy than the DFWCS, and probably requires at least three independent failures to cause a system failure. It is expected that sequences/cutsets of orders higher than 3 will have to be considered, and a much larger number of sequences evaluated. Therefore, failure modes may have to be grouped and repair be considered at a higher level of detail, as discussed previously.

- It is important that a reliability model realistically captures the fault-tolerance features of a digital system. This is often accounted for by adjusting component failure rates or probabilities with fault coverage values (i.e., the fraction of faults that would be

automatically detected and compensated for). In this study, for each failure mode associated with a CPU module which has an independent WDT, plant information and an understanding about how the system works were used to determine if the effect of each failure mode on the module can be detected by its WDT and/or the application software. The probability that an individual failure mode or sequence is detected by the WDT was assumed to be either one or zero given that the WDT functions properly. In this sense, the coverage is automatically accounted for in the probabilities of all failure sequences. However, due to limitations in the state of the art for FMEA, whether or not the failure modes of some components, such as a random access memory (RAM), can be detected by the fault tolerance features was determined subjectively. The concept of fault coverage can be used to improve this treatment. In general, fault coverage can be used to adjust the component failure rates, as in Aldemir [2009], which estimated coverages using fault injection experiments. If fault coverage is accounted for in the failure data, then detailed models of the fault-tolerance features do not have to be explicitly included in the reliability models. Coverage of fault-tolerance features is an area for future research.

- An important assumption of the Markov model described in Chapter 5 is that online repair is not possible, which is the case for the DFWCS. For other digital systems, such as an RPS, on-line repair may be possible, and the analytical solutions of the Markov model developed in Chapter 5 cannot be used. If components can be repaired with the system operating, the Markov model would have to be modified by adding transitions that represent repairs, making it much more difficult to solve. Using the simplified Markov model described in Section 5.3, the governing equations in the Laplace-transformed space can be solved analytically, and the inverse Laplace transform can be solved in the same way described in Section 5.3. The accuracy of the simplified Markov method needs to be further explored and, if necessary, better approximate methods can be developed. Alternatively, as discussed in the previous insight, it may be possible to develop a higher level model based on the more detailed model and, as discussed in Section 5.4, numerically solve the higher level model even if it includes repair.

- The proposed FMEA approach and its implementation assume only one failure mode for some components, such as the Industry Standard Architecture bus, RAM, Read-Only Memory, Basic Input/Output System, flash disk, serial port, address logic, and buffer. The only failure mode for these components is the loss of the component. In many, but not all, cases these were considered to be undetectable failures because of the difficulty in precisely evaluating their impacts. The automated FMEA tool can be enhanced by defining more detailed failure modes for these components. For example, some of the lower level failure modes of RAM may be detectable, while some other failure modes are not. This is an issue that can be addressed using the concept of coverage, as discussed above and in Section 3.4. While a more systematic treatment of the detectability of component failure modes is desirable, it should also be recognized that detectability of a failure mode is design specific and coverage values obtained for one system will often not be applicable to other systems.

- The quantification method used in this study can estimate the upper bound of errors due to truncation based on the order of the failure sequences, and this upper bound can be used in determining if convergence has been achieved.

11.3 Areas of Potential Additional Research

The experience of developing the probabilistic model of the DFWCS identified many areas of research to enhance the state of the art in modeling digital systems. They have been discussed throughout the report and are summarized below.

- Improved approaches for defining and identifying failure modes of digital systems should be developed. Both software and hardware failure modes need to be considered. In this study, generic component failure modes that are publicly available are used. As discussed in Chapter 3, the component failure modes may not be complete, and the breakdown of component failure rates into constituent failure modes may not be supported by adequate failure data. Software failures are beyond the scope of this study, and placeholders for two generic software failure modes are used in the model of the DFWCS. Research on software failure modes that can be incorporated in reliability models of digital systems is needed. A review of software failure experience in different industries would be beneficial. Also, there are unique features in a digital design, such as communication and synchronization, whose failure modes and effects are not well understood and may introduce dependencies between redundant equipment. Therefore, more research is needed to evaluate the potential failure modes and effects associated with these features.

- Software reliability methods for quantifying the likelihood of failures of both application and support software need to be developed. Many risk analysts believe that software CCFs are the most risk significant failures for digital I&C systems. However, it is difficult to determine how significant the impacts may be without quantifying them. Also, methods for modeling software CCFs across system boundaries (e.g., due to common support software) need to be developed, as suggested in 4.6.4 of Section 9.4.

- Methods and parameter data for modeling self-diagnostics, reconfiguration, and surveillance, including using other components to detect failures, are needed. In this study, the automated FMEA tool captures the fault-tolerance features implemented in the application software and by the WDTs. Fault-tolerance features are not limited to those modeled in this study. Different hardware redundancy techniques and software fault-tolerance designs can be applied to digital system designs. Incorporation of these different designs needs to be further pursued.

- Chapter 6 discusses how publicly available hardware failure data are used in this study and points out that better data for hardware failures and a break down of the failure rates by failure modes of digital components need to be collected. The potential issue of double-crediting fault-tolerant features, such as self-diagnostics[11], discussed in NUREG/CR-6962 [Chu 2008a], needs to be addressed. The research should include collection and analysis of generic manufacturer data and specific operating data.

- Better data for the CCFs of digital components need to be collected. The reason for using 0.05 as the beta factor in this study is a lack of applicable data and should not be considered conservative. It is acknowledged that CCF data for digital components are sparse and further investigation is needed in this area.

[11] Double-crediting fault-tolerant features also can be an issue for software failures.

- Use of Markov quantification methods raises some issues with regard to integration with a PRA that is based on the ET/FT method. Integration of Markov models, such as the one developed in this study, with an ET/FT PRA should be demonstrated.

- Methods for human reliability analysis (HRA) associated with digital systems need to be investigated. In this study, a loss of automatic control of the DFWCS is defined as a system failure. It should be recognized that operator action may still be able to maintain the feedwater level manually without causing an initiating event. In addition, different failure modes may generate different alarms and/or annunciations, which are likely to affect performance of the operator in different ways. Additional research in this area would help create more realistic reliability models of digital systems. In general, digital upgrades at current nuclear power plants and the designs of new reactors introduce new human system interfaces that are significantly different from those of existing plants. HRA research is needed to address these new interfaces in support of PRAs for both existing plants and new reactors.

- This study identified that it may be beneficial to include controlled processes in modeling drifting signals of a control system, but not necessarily for a protection system. It is also not clear whether the increased accuracy of modeling obtained through incorporation of a plant dynamics model would justify the increased complexity and effort required for intensive simulation. Determining if and when a model of controlled processes is necessary in developing a reliability model of a digital system should be further researched.

12. REFERENCES

Aeroflex, "Reliability Failure Mode Effects and Predicted Failure Rate Analysis for the ACT8500 64-Channel Multiplexer Module," Application Note AN8500-1, September 15, 2005.

Aldemir, T., Miller, D. W., Stovsky, M., Kirschenbaum, J., Bucci, P., Fentiman, A. W., and Mangan, L. M., "Current State of Reliability Modeling Methodologies for Digital Systems and Their Acceptance Criteria for Nuclear Power Plant Assessments," NUREG/CR-6901, February 2006.

Aldemir, T., Stovsky, M. P., Kirschenbaum, J., Mandelli, D., Bucci, P., Mangan, L. A., Miller, D. W., Sun, X., Ekici, E., Guarro, S., Yau, M., Johnson, B. W., Elks, C., and Arndt, S. A., "Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments," NUREG/CR-6942, October 2007.

Aldemir, T., Gurro, S., Kirschenbaum, J., Mandelli, D., Mangan, L.A., Bucci, P., Yau, M., Johnson, B., Elks, C., Ekici, E., Stovsky, M.P., Miller, D.W., Sun, X., Arndt, S.A., Nguyen, Q., and Dion, J., "A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems," NUREG/CR-6985, February 2009.

Apostolakis, G.E., and Chu, T.L., "Unavailability of Systems Under Periodic Test and Maintenance," *Nuclear Technology*, v 50, n 1, Mid-Aug, p 5-15, 1980.

Apostolakis, G. and Kaplan, S., "Pitfalls in Risk Calculations," *Reliability Engineering*, v 2, n 2, p. 135-145, April-June 1981.

Atwood, C., "Handbook of Parameter Estimation for Probabilistic Risk Assessment," NUREG/CR-6823, September 2003.

Blanton, C.H., and Eide, S.A., "Savannah River Site Generic Data Base Development (U)," WSRC-TR-93-262, Westinghouse Savannah River Company, June 1993.

Chu, T.L., Martinez-Guridi, G., Lehner, J., and Overland, D., "Issues Associated with Probabilistic Failure Modeling of Digital Systems," NPIC & HMIT 2004, Columbus, Ohio, September 2004.

Chu, T. L., Martinez-Guridi, G., Yue, M., and J. Lehner, "A Review of Software-Induced Failure Experience," NPIC & HMIT 2006, Albuquerque, New Mexico, November 12-16, 2006.

Chu, T.L., Martinez-Guridi, G., Yue, M., and J. Lehner, "Basis for Using Software Failure Rates and Probabilities in Probabilistic Failure Modeling of Digital Systems of a Nuclear Power Plant," IAEA Technical Meeting on Common Cause Failures in Digital Instrumentation and Control Systems of Nuclear Power Plants, Bethesda, Maryland, USA, June 19-21, 2007.

Chu, T.L., Martinez-Guridi, G., Yue, M., Lehner, J., and Samanta, P., "Traditional Probabilistic Risk Assessment Methods for Digital Systems," NUREG/CR-6962, July 2008a.

Chu, T.L., and Yue, M., "Analytical Calculation of the Mean Value of a Top Event," PSA 2008, September 7 – 11, 2008, Knoxville, Tennessee, 2008b.

Department of Defense, “Reliability Prediction of Electronic Equipment,” Notice 2, MIL-HDBK-217F, February 18, 1995.

Elks, C., et al., “Quantitative Dependability Assessment of the Benchmark Digital Feed-Water Control System: Final Report,” University of Virginia, Technical Report UVA-CSCS 2007-003, January 20, 2008.

Electric Power Research Institute, ALWR Utility Requirements Document, Vol. II, ALWR Evolutionary Plant, Ch. 1, App. A, PRA Key Assumptions and Ground Rules, Rev. 6, December 1993.

Eurotherm Ltd., Using 2604/2704 Fixed Digital I/O, Technical Information, No. TIN 137, pg. B-113, 2000.

Garrett, C., and Apostolakis, G., “Context in the Risk Assessment of Digital Systems,” *Risk Analysis*, Vol. 19, n 1, p. 23-32, February 1999.

Gu, J., and Pecht, M., “Predicting the Reliability of Electronic Products,” 8th International Conference on Electronic Packaging Technology, Shanghai, China, August 2007.

Kaplan, S., “On a Two-stage Bayesian Procedure for Determining Failure Rates,” *Institute of Electrical and Electronics Engineers Transactions on Reliability*, R-33, 227-232, 1984.

Meeldijk, V., *Electronic Components Selection and Application Guidelines*, John Wiley & Sons, 1996.

Nuclear Energy Agency, “Operation and Maintenance Experience with Computer-Based Systems in Nuclear Power Plants,” Committee on the Safety of Nuclear Installations, A Report by the PWR-1 Task Group on Computer-based Systems Important to Safety, NEA/CSNI/R(97)23, September 10, 1998.

Nuclear Regulatory Commission, “NRC Digital System Research Plan, FY 2005 − FY 2009,” Revision 06/2, April 2006.

Nuclear Regulatory Commission Policy Statement, “Use of Probabilistic Risk Assessment Methods in Nuclear Regulatory Activities,” August 16, 1995.

Pecht, M.G., and Nash, F.R., “Predicting the Reliability of Electronic Equipment,“ *Proceedings of the Institute of Electrical and Electronics Engineers*, Vol. 82, No. 7, July 1994.

Reliability Analysis Center, “Failure Mode/Mechanism Distributions,” A Department of Defense Information Analysis Center, FMD-97, December 1997b.

Reliability Analysis Center, “PRISM Users Manual, Version 1.4,” Prepared by Reliability Analysis Center Under Contract to Defense Supply Center Columbus.

Rouvroye, J.L., and Brombacher, A.C., “New Quantitative Safety Standards: Different Techniques, Different Results,” *Reliability Engineering and System Safety*, v 66, n 2, p. 121-125, November 1999.

Telcordia, ”Reliability Prediction Procedure for Electronic Equipment,” SR-332 Issue 1, May 2001.

Wierman, T.E., et al. “Reliability Study: Combustion Engineering Reactor Protection System, 1984 -1998,” NUREG/CR-5500, Vol. 10, 2002.

Yue, M., and Chu, T.L., “Estimation of Failure Rates of Digital Components Using a Hierarchical Bayesian Method,” PSAM8, New Orleans, Louisiana, May 14-19, 2006.