

To the Graduate Council:

I am submitting herewith a dissertation written by Fan Li entitled “Dynamic Modeling, Sensor Placement Design, and Fault Diagnosis of Nuclear Desalination Systems.” I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Nuclear Engineering.

Belle R. Upadhyaya, Major Professor

We have read this dissertation
and recommend its acceptance:

J. Wesley Hines

Haitao Liao

Rao V. Arimilli

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Dynamic Modeling, Sensor Placement Design, and Fault Diagnosis of Nuclear Desalination Systems

A Dissertation Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Fan Li
May 2011

Copyright © 2011 by Fan Li
All rights reserved.

Dedication

This dissertation is dedicated to my family in China for always being supportive in my pursuit of higher education and professional development. I also dedicate this degree to my dear wife, Xiaojia Xu, for her understanding, encouragement, and help for the past ten years.

Acknowledgments

I would like to thank all those who have helped me in completing my doctorate degree in Nuclear Engineering at the University of Tennessee, Knoxville. I am especially grateful to Dr. Belle R. Upadhyaya, my academic advisor, for his guidance, advice, inspiration, and valuable suggestions during the course of this dissertation. Sincere thanks also go to the members of my doctoral committee, Dr. Rao V. Arimilli, Dr. J. Wesley Hines, and Dr. Haitao Liao, for their comments and suggestions. In addition, I am deeply thankful to the head of the Nuclear Engineering Department, Dr. H. Lee Dodds, for providing me with the precious opportunity to study and work at UTNE.

Also, I would like to thank my colleagues, Dr. Sergio R.P. Perillo and Ms. Xiaojia Xu, for many technical discussions throughout my graduate studies at UTNE.

This research is supported by a U.S. Department of Energy NERI-C grant with the University of Tennessee, Knoxville.

Abstract

Fault diagnosis of sensors, devices, and equipment is an important topic in the nuclear industry for effective and continuous operation of nuclear power plants. All the fault diagnostic approaches depend critically on the sensors that measure important process variables. Whenever a process encounters a fault, the effect of the fault is propagated to some or all the process variables. The ability of the sensor network to detect and isolate failure modes and anomalous conditions is crucial for the effectiveness of a fault detection and isolation (FDI) system. However, the emphasis of most fault diagnostic approaches found in the literature is primarily on the procedures for performing FDI using a given set of sensors. Little attention has been given to actual sensor allocation for achieving the efficient FDI performance. This dissertation presents a graph-based approach that serves as a solution for the optimization of sensor placement to ensure the observability of faults, as well as the fault resolution to a maximum possible extent. This would potentially facilitate an automated sensor allocation procedure. Principal component analysis (PCA), a multivariate data-driven technique, is used to capture the relationships in the data, and to fit a hyper-plane to the data. The fault directions for different fault scenarios are obtained from the prediction errors, and fault isolation is then accomplished using new projections on these fault directions. The effectiveness of the use of an optimal sensor set versus a reduced set for fault detection and isolation is demonstrated using this technique.

Among a variety of desalination technologies, the multi-stage flash (MSF) processes contribute substantially to the desalinating capacity in the world. In this dissertation, both steady-state and dynamic simulation models of a MSF desalination plant are developed. The dynamic MSF model is coupled with a previously developed International Reactor Innovative and Secure (IRIS) model in the SIMULINK environment. The developed sensor placement design and fault diagnostic methods are illustrated with application to the coupled nuclear desalination system. The results demonstrate the effectiveness of the newly developed integrated approach to performance monitoring and fault diagnosis with optimized sensor placement for large industrial systems.

Table of Contents

Chapter 1	1
Introduction	1
1.1 Background and Motivation	1
1.2 R&D Objectives and Original Contributions	5
1.3 Organization of the Dissertation	7
Chapter 2	8
Literature Review	8
2.1 Optimum Sensor Placement for Process Systems	8
2.1.1 Modeling of Systems	9
2.1.2 Sensor Selection Evaluation	16
2.1.3 Sensor Optimization Techniques	22
2.2 Seawater Desalination Methods	24
2.2.1 Classification of Desalination Processes	25
2.2.2 Distillation Desalination Systems	26
2.2.3 Membrane Desalination Systems	31
Chapter 3	35
Integrated Architecture of a Fault Diagnostic System	35
Chapter 4	38
Modeling and Simulation of a Multi-Stage Flash Desalination System	38
4.1 Introduction	38
4.2 MSF Desalination System Description	38
4.3 MSF Mathematical Models	41
4.3.1 Flashing Stage Dynamic Model	41
4.3.2 Brine Heater Dynamic Model	44
4.3.3 MSF Steady-State Model	44
4.4 Control Loops in a MSF System	46
4.5 Measurements and Instrumentation in a MSF System	49
4.6 MSF Desalination System Simulation Results	51
4.6.1 Steady-State Simulation Results	51
4.6.2 Dynamic Simulation Results	56
4.7 Summary	63
Chapter 5	66
Modeling and Simulation of a Nuclear Desalination Plant	66
5.1 Introduction	66
5.2 IRIS System Description	66
5.3 IRIS Dynamic Models	71
5.3.1 Reactor Core	71
5.3.2 Helical Coil Steam Generator	71
5.3.3 Steam Turbine	72
5.3.4 Main Condenser	73
5.3.5 Feedwater Heater	74
5.4 IRIS-MSF Nuclear Desalination Plant Simulation Results	75

5.5	Summary	85
Chapter 6		86
Development of Sensor Placement Design Algorithms.....		86
6.1	Introduction	86
6.2	Methodology of Directed Graph Modeling	87
6.3	Sensor Location and Fault Diagnostic Observability Criterion	89
6.3.1	Set-Covering Problems	89
6.3.2	Fault Observability.....	91
6.3.3	Fault Resolution	93
6.4	Sensor Location and System Unobservability Criterion	96
6.4.1	Formulation of Unobservability Minimization	96
6.4.2	One-Step Optimization Formulation.....	99
6.4.3	Solving Unobservability Minimization Problems	100
6.5	Summary	104
Chapter 7		105
Sensor Placement Design and Fault Diagnosis Application for a MSF Desalination System ...		105
7.1	Introduction	105
7.2	Principal Component Analysis for Fault Diagnosis	105
7.2.1	PCA Algorithm	106
7.2.2	Fault Detection.....	109
7.2.3	Fault Isolation	110
7.3	DG Model Development of a MSF System	112
7.4	Sensor Placement Design	120
7.5	PCA Model Development of a MSF System	122
7.5.1	Database Generation	122
7.5.2	PCA Modeling	122
7.6	Fault Detection Results	125
7.7	Fault Isolation Results	130
7.7.1	Single-Fault Cases	130
7.7.2	Dual-Fault Cases	135
7.8	Summary	142
Chapter 8		143
Sensor Placement Design and Fault Diagnosis Application for HCSG Systems		143
8.1	HCSG System Description	143
8.2	Sensor Placement Design	147
8.2.1	Fault Diagnostic Observability Criterion.....	147
8.2.2	System Unobservability Criterion.....	147
8.3	PCA Fault Diagnostic Results	151
8.4	Summary	159
Chapter 9		164
Summary, Conclusions, and Suggestions for Future Research		164
9.1	Summary and Conclusions	164
9.2	Suggestions for Future Research	168
List of References		169
Appendices.....		188

Appendix A:	Thermo-Physical Properties	189
Appendix B:	IRIS Nuclear System Mathematical Model.....	193
Appendix C:	Fault Isolation Index Plots of Dual-Fault Scenarios	204
Appendix D:	MSF Desalination Plant Simulation Codes.....	215
Appendix E:	IRIS-MSF Nuclear Desalination Simulation Codes.....	225
Appendix F:	Greedy Search Optimization Algorithm Codes	235
Appendix G:	PCA Fault Detection and Isolation Codes	243
Vita.....		248

List of Tables

Table 4.1. Measuring instruments used in the desalination industry	50
Table 4.2. The main characteristics of the plant	53
Table 4.3. The operating conditions	53
Table 4.4. The condenser tube bundle data.....	53
Table 4.5. Steady-state model results corresponding to the performance study case.....	54
Table 4.6. Actual plant operation data [Helal et al., 1986]	55
Table 4.7. Control loops in dynamic simulation.....	57
Table 4.8. Dynamic simulation tests.....	58
Table 6.1. Set A of sensors for different fault nodes in Figure 6.2.....	96
Table 6.2. Set B_{ij} for single-fault case of Table 6.1	96
Table 7.1. DG model variables of brine heater.....	115
Table 7.2. DG model variables of flash stage i	117
Table 7.3. Two-stage MSF system available variables.....	119
Table 7.4. Faults considered for two-stage MSF system.....	120
Table 7.5. MSF variables used to develop PCA models.....	123
Table 7.6. PCA fault direction analysis for MSF system under single fault assumption ...	130
Table 8.1. Fault nodes of a pair of HCSGs	146
Table 8.2. Fault data for a pair of HCSGs	148
Table 8.3. Sensor data for a pair of HCSGs.....	148
Table 8.4. Sensor selection results using greedy search heuristic and LINGO 8.0	149
Table 8.5. HCSG sensor placement results ($TC=1000$).....	150
Table 8.6. PCA fault direction analysis for a pair of HCSGs.....	155

List of Figures

Figure 1.1. Projected regions affected by water shortages [IWMI, 2000].	1
Figure 2.1. Basic components of sensor placement strategy.	10
Figure 2.2. Directed graph of a hypothetical system.	21
Figure 2.3. Particle swarm optimization.	24
Figure 2.4. Classification of desalination processes [IAEA, 2007].	25
Figure 2.5. Principle of vapor compression [Khan, 1986].	27
Figure 2.6. Schematic diagram of a MED system [IAEA, 2007].	28
Figure 2.7. Operating principle of a single stage MSF [IAEA, 2007].	30
Figure 2.8. Schematic diagram of an industrial MSF design [IAEA, 2007].	30
Figure 2.9. Osmosis and reverse osmosis processes [IAEA, 2007].	32
Figure 2.10. Principle of electro dialysis [Khan, 1986].	34
Figure 3.1. An integrated architecture of a fault diagnostic system.	37
Figure 4.1. Schematic diagram of a MSF process with brine recirculation.	40
Figure 4.2. Block diagram of a generic MSF stage.	42
Figure 4.3. Single-input single-output MSF control loops [Alatqi et al., 1999].	46
Figure 4.4. Screenshot of MSF SIMULINK models.	57
Figure 4.5. Dynamic responses of process variables to step increase of TBT set point.	59
Figure 4.6. Dynamic responses of process variables to step increase of incoming cooling brine flow rate.	60
Figure 4.7. Dynamic responses of process variables to step increase of incoming cooling brine temperature.	61
Figure 4.8. Dynamic responses of process variables to step increase of heating steam temperature.	62
Figure 4.9. Open loop response of TBT to step decrease of heating steam temperature.	64
Figure 4.10. Open loop response of brine levels to step decrease of heating steam temperature.	64
Figure 4.11. Open loop response of last-stage brine level to step decrease of heating steam temperature.	65
Figure 5.1. IRIS primary system layout [Carelli et al., 2003].	69
Figure 5.2. IRIS reactor pressure vessel and compact containment vessel [Carelli et al., 2003].	70
Figure 5.3. Schematic diagram of the turbine condenser model.	74
Figure 5.4. Schematic of a horizontal closed feedwater heater [Thanomsat et al., 1998].	75
Figure 5.5. Schematic diagram of a MSF plant coupled to the IRIS power conversion system.	76
Figure 5.6. Reactor power response to a 10% power demand step decrease.	78
Figure 5.7. T_{hot} response to a 10% power demand step decrease.	78
Figure 5.8. T_{cold} response to a 10% power demand step decrease.	79
Figure 5.9. T_{ave} response to a 10% power demand step decrease.	79
Figure 5.10. Steam pressure response to a 10% power demand step decrease.	80
Figure 5.11. Steam flow rate response to a 10% power demand step decrease.	80

Figure 5.12. Turbine output response to a 10% power demand step decrease.	81
Figure 5.13. Feedwater temperature response to a 10% power demand step decrease.	81
Figure 5.14. Steam temperature response to a 10% power demand step decrease.	82
Figure 5.15. Top brine temperature response to a 10% power demand step decrease.	82
Figure 5.16. Brine heater inlet steam flow rate response to a 10% power demand step decrease.	83
Figure 5.17. Brine heater inlet steam enthalpy response to a 10% power demand step decrease.	83
Figure 5.18. Brine level response to a 10% power demand step decrease.	84
Figure 6.1. Directed graph of a hypothetical process.	88
Figure 6.2. Fault and sensor node bipartite graph for Figure 6.1.	93
Figure 6.3. Flowchart for greedy search algorithm.	101
Figure 6.4. Bipartite graph for Figure 6.2 with probabilities.	103
Figure 7.1. Brine heater section of a MSF plant.	113
Figure 7.2. Directed graph of brine heater section.	114
Figure 7.3. A generic flash stage of a MSF plant.	116
Figure 7.4. Directed graph of a flash stage i	116
Figure 7.5. Directed graph of a two-stage MSF system.	118
Figure 7.6. Percentage of variance explained by principal components.	123
Figure 7.7. Q statistic for MSF PCA model.	124
Figure 7.8. T^2 statistic for MSF PCA model.	124
Figure 7.9. Residual pattern for inlet cooling brine temperature sensor drift (Fault #1). ...	126
Figure 7.10. Residual pattern for inlet cooling brine flow sensor drift (Fault #2).	126
Figure 7.11. Residual pattern for inlet cooling brine salinity sensor drift (Fault #3).	127
Figure 7.12. Residual pattern for flashing stage #1 heat transfer degradation (Fault #4)...	127
Figure 7.13. Residual pattern for flashing stage #2 heat transfer degradation (Fault #5)...	128
Figure 7.14. Residual pattern for top brine temperature sensor drift (Fault #6).	128
Figure 7.15. Residual pattern for top brine temperature controller fault (Fault #7).	129
Figure 7.16. Residual pattern for Stage #2 brine level controller fault (Fault #8).	129
Figure 7.17. Fault isolation index for inlet cooling brine temperature sensor drift.	131
Figure 7.18. Fault isolation index for inlet cooling brine flow sensor drift.	131
Figure 7.19. Fault isolation index for inlet cooling brine salinity sensor drift.	132
Figure 7.20. Fault isolation index for flashing stage #1 heat transfer degradation.	132
Figure 7.21. Fault isolation index for flashing stage #2 heat transfer degradation.	133
Figure 7.22. Fault isolation index for top brine temperature sensor drift.	133
Figure 7.23. Fault isolation index for top brine temperature controller fault.	134
Figure 7.24. Fault isolation index for Stage #2 brine level controller fault.	134
Figure 7.25 Fault isolation index for dual-fault case #1.	139
Figure 7.26 Fault isolation index for dual-fault case #2.	139
Figure 7.27 Fault isolation index for dual-fault case #3.	140
Figure 7.28 Fault isolation index for dual-fault case #4.	140
Figure 7.29 Fault isolation index for dual-fault case #5.	141
Figure 7.30 Fault isolation index for dual-fault case #6.	141
Figure 7.31 Fault isolation index for dual-fault case #7.	142
Figure 8.1. IRIS steam generator layout [Carelli et al., 2003].	143

Figure 8.2. A schematic of a pair of HCSGs [Zhao, 2005a].....	144
Figure 8.3. Directed graph of a pair of HCSGs.	146
Figure 8.4. Unobservability trending of the faults in a pair of HCSGs.	150
Figure 8.5. Residual pattern for T_{hot} sensor fault (Fault #1).	152
Figure 8.6. Residual pattern for SG-A thermal degradation (Fault #2).	152
Figure 8.7. Residual pattern for SG-B thermal degradation (Fault #3).	153
Figure 8.8. Residual pattern for secondary flow distribution anomaly (Fault #4).	153
Figure 8.9. Residual pattern for feed temperature sensor fault (Fault #5).	154
Figure 8.10. Residual pattern for feed flow sensor fault (Fault #6).	154
Figure 8.11. Fault isolation index for T_{hot} sensor fault.	156
Figure 8.12. Fault isolation index for SG-A thermal degradation.	156
Figure 8.13. Fault isolation index for SG-B thermal degradation.	157
Figure 8.14. Fault isolation index for secondary flow distribution anomaly.	157
Figure 8.15. Fault isolation index for feed temperature sensor fault.	158
Figure 8.16. Fault isolation index for feed flow sensor fault.	158
Figure 8.17. Fault isolation index for T_{hot} sensor fault (reduced sensor set).	160
Figure 8.18. Fault isolation index for SG-A thermal degradation (reduced sensor set).	160
Figure 8.19. Fault isolation index for SG-B thermal degradation (reduced sensor set).	161
Figure 8.20. Fault isolation index for secondary flow anomaly (reduced sensor set).	161
Figure 8.21. Fault isolation index for feed temperature sensor fault (reduced sensor set).	162
Figure 8.22. Fault isolation index for feed flow sensor fault (reduced sensor set).	162
Figure 8.23. Fault isolation index for a pair of HCSGs.	163

Abbreviations and Symbols

AAKR	Auto-Associative Kernel Regression
AIC	Akaike Information Criterion
ANN	Artificial Neural Network
BOP	Balance-of-Plant
BPE	Boiling Point Elevation
CBM	Condition-Based Maintenance
CPV	Cumulative Percentage Variance
CRDM	Control Rod Drive Mechanism
CV	Control Volume
DG	Directed Graph
ED	Electro-Dialysis
FDI	Fault Detection and Isolation
FOM	Figures of Merit
GA	Genetic Algorithm
GMDH	Group Method of Data Handling
GUI	Graphical User Interface
HCSG	Helical Coil Steam Generator
HTE	Horizontal Tube Evaporator
HT-VTE	High Temperature Vertical Tube Evaporation
IAEA	International Atomic Energy Agency
ILP	Integer Linear Programming
INLP	Integer Non-Linear Programming
IPSR	Integral Primary System Reactor
IRIS	International Reactor Innovative and Secure
IWMI	International Water Management Institute
LMTD	Log Mean Temperature Difference
LOCA	Loss-of-Coolant Accident
LT-HTME	Low Temperature Horizontal Tube Multi-Effect

LWR	Light Water Reactor
MED	Multi-Effect Distillation
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MSF	Multi-Stage Flash
NEA	Non-Equilibrium Allowance
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
PWR	Pressurized Water Reactor
QDG	Quantified Directed Graph
RCP	Reactor Coolant Pump
RO	Reverse Osmosis
SA	Simulated Annealing
SMR	Small Modular Reactor
SNR	Signal-to-Noise Ratio
SPE	Squared Prediction Error
SVD	Singular Value Decomposition
TBT	Top Brine Temperature
TDM	Tri-Diagonal Matrix
VC	Vapor Compression
VRE	Variance Reconstruction Error

Chapter 1

Introduction

1.1 Background and Motivation

Water is indispensable for the very existence of mankind and for human development. The total quantity of water available on earth is about 1,000 million km³ and covers nearly 70% of the globe, whereas the total world water consumption does not exceed 2,100 km³/year. At first thought this would seem rather reassuring. However, 97.5% of the available water is highly saline or brackish. Of the remaining 2.5%, nearly 70% is in the form of ice (Antarctica, Greenland, etc). Yet another large fraction is locked in the soil humidity and in deep underground aquifers. Consequently the effective amount of water, directly accessible to human beings is only 0.007% (or, about 70,000 km³) [IWMI, 2000]. Even this fraction is very unevenly distributed over the planet. Moreover, rapidly increasing populations, rising standards of living, continued development of tourism, progressive industrialization, and expansion of irrigation agriculture have already led to acute water shortages and stresses in many regions of the world as shown in red in Figure 1.1.

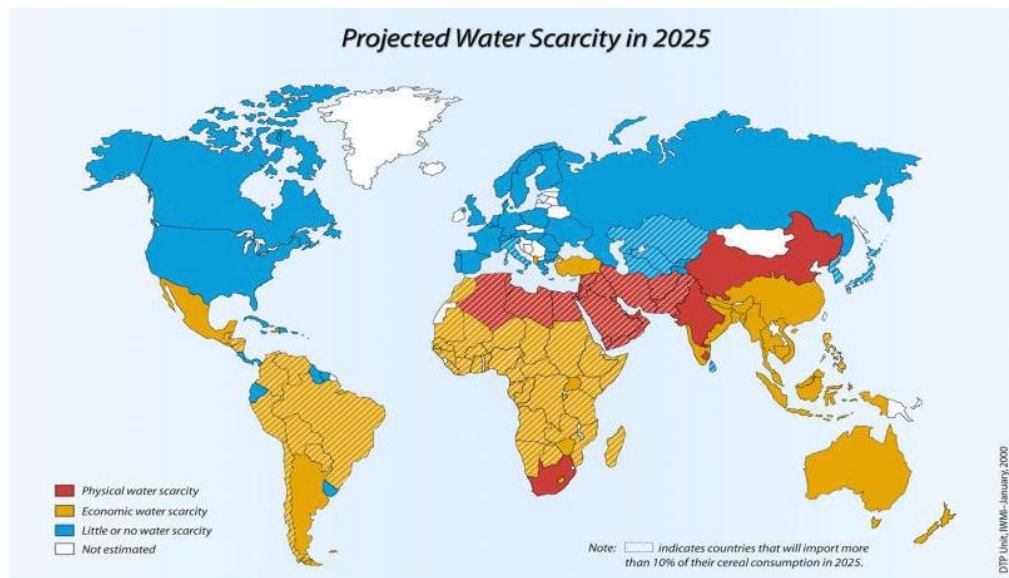


Figure 1.1. Projected regions affected by water shortages [IWMI, 2000].

In Figure 1.1, countries which will face “economic water shortages” (i.e. inadequacy of supply and demand) are shown. According to the market survey performed by the World Resources Institute on the future growth of seawater desalination, the worldwide demand for desalination is expected to double approximately every 10 years in the foreseeable future. Most of the demand would arise in the Arabian Gulf and North African regions, but this is likely to expand to other areas.

During the second half of the past century, industrial water desalination became a sustainable source of potable water in several countries across the globe. Out of the more than 12,000 desalination plants currently in operation, only about 10 use heat or electricity provided by nuclear power plants [IAEA, 2002]. Fossil energy sources are the dominant choice. However, due to their constantly depleting quantities and the emitted air pollutants as a result of combustion, fossil fuels should not be considered as a sustainable source of energy. These environmental concerns, coupled with concerns over energy supply security and an anticipated growth in energy demands, are driving a growing interest in the development and expansion of the nuclear energy options. Nuclear energy offers a clean and abundant energy supply. Also, the current generation of nuclear plants has proven that nuclear energy can be safe and economically competitive with alternative options.

In the International Atomic Energy Agency (IAEA) terminology, nuclear desalination is defined as *the production of potable water from seawater in a facility in which a nuclear reactor is used as the source of energy for the desalination process. Electrical and/or thermal energy may be used in the desalination process. The facility may be dedicated solely to the production of potable water, or may be used for the generation of electricity and production of potable water, in which case only a portion of the total energy output of the reactor is used for water production* [IAEA, 2007]. The prospects of using nuclear energy for seawater desalination on a large scale can be very attractive since desalination is an energy intensive process that can utilize the heat from a nuclear reactor and/or the electricity produced by such plants. The choice of the desalination technology determines the form of energy required: electrical energy for reverse osmosis (RO) systems, and relatively low quality thermal energy for distillation systems, such as multi-stage flash (MSF) distillation. The MSF desalination process has dominated the field of

thermal desalination with a market share of over 90%. In addition, market share of MSF processes accounts for 70% of all seawater desalination processes [El-Dessouky et al., 1999]. It is thus logical that the capability of coupling a MSF process to a nuclear power plant as the heat source makes the process increasingly important for future drinking water and electricity generation.

On the other hand, while all nuclear reactor types can provide the energy required by the various desalination processes, it has been shown that small and medium sized reactors (SMR), ranging from 50 MWe to 500 MWe in electricity production, offer the largest potential as the coupling option of nuclear desalination systems in developing countries. The development of innovative reactor concepts and fuel cycles with enhanced safety features as well as their attractive economics are expected to improve the public acceptance and further the prospects of nuclear desalination.

To facilitate the anticipated growth in demand for nuclear energy world-wide, several countries including the United States, have initiated the development of the next generation of nuclear plants that offer even greater safety, reliability, and economics, while also reducing the threat of proliferation of special nuclear materials. The Generation IV nuclear power program was initiated by the U.S. Department of Energy to identify and develop promising next-generation nuclear plant designs, and was expanded to include several other countries. Six advanced reactor designs were selected by the international Generation IV program in 2003 for long-term development and several designs that are viewed as near-term deployable by 2015. One of the reactor types identified for international near-term deployment is integral primary system reactors (IPSR). An example of an IPSR system is the International Reactor Innovative and Secure (IRIS) reactor concept [Carelli et al., 2003]. IRIS is a modular design of a pressurized water reactor (PWR) that has an integral reactor coolant system with enhanced safety and economics. IRIS is especially well suited for deployment in countries with small or medium electricity grids for producing both electricity and fresh water. However, coupling a nuclear plant and a desalination plant involves a number of issues that have to be addressed. Among these issues, performance monitoring and fault diagnosis are of high importance for safe and optimal operation of a coupled nuclear desalination plant.

The main objective of fault diagnosis is to observe incipient faults and determine the root causes. Nowadays, most processes are extremely well instrumented with a multitude of sensors providing both control and safety-related measurements, especially in a nuclear desalination plant. The faults with control-related sensors can propagate the fault effects through control loops and subsequently disturb process variables throughout the entire process, which will result in a non-optimal performance of the related actuators and plant equipment. The safety-related sensor faults can force a nuclear power plant to de-rate its power and degrade the operational performance. Both types of sensor faults can also mislead operators to take erroneous actions and in turn result in safety problems. The ability of the sensor network to detect and discriminate failure modes and anomalous conditions is critical for the effectiveness of the fault detection and isolation (FDI) system. With hundreds of process variables available for measurement in both nuclear power plants and desalination processes, selection of an optimum number of sensors and their locations poses a unique challenge.

FDI has been considered as an important design feature of the advanced instrumentation and control systems of the IRIS design. However, conventional sensor FDI techniques, such as hardware redundancy, face many challenges in engineering applications. As an integral reactor, IRIS has the entire reactor coolant system housed inside the reactor vessel. This arrangement creates engineering difficulties in placing physically redundant instruments for in-vessel equipment due to the need for extra space and instrument penetrations. In addition, the possible common-cause failures of physically redundant sensors may still constrain the reliability of the instrument system.

Many modern FDI approaches are based on analytical redundancy. Functional relationships among process variables governed by fundamental conservation laws such as mass, momentum, and energy balance, can replace hardware redundancy for plant measurements [Chow et al., 1984; Erbay and Upadhyaya, 1997; Holbert and Upadhyaya, 1990; Upadhyaya et al., 1989; Upadhyaya and Eryurek, 1992]. Residuals, defined as the differences between measured values and estimated values, can be continuously monitored for fault detection. Because different faults cause the violation of different relationships, residual patterns can be used as fault signatures for fault isolation.

Because of the complexity of either a nuclear power plant or a seawater desalination plant, it is generally very difficult to build physics models with sufficient accuracy for FDI purpose. Thus, the so-called soft computing methods, such as artificial neural networks (ANN) [Kavaklioglu et al., 1994], principal component analysis (PCA) [Wang et al., 2002], fuzzy logic [Zhao et al., 2005], group method of data handling (GMDH) [Farlow, 1984], and other data-based empirical modeling techniques [Hines et al., 1997; Hines et al., 2008], have shown great capabilities of capturing the relationships among the various measurements.

Multivariate statistical process control using empirical techniques, such as PCA for data characterization, has been employed successfully in the process industries. In chemical industry, PCA has been one of the most popular statistical methods for extracting information from process data. Dunia and Qin [1998] proposed a subspace approach for fault detection, reconstruction, isolation, and identification. Due to the difficulty of isolating faults from process parameter changes when the Hotelling's T^2 test is violated, their method was only based on the Q statistic, or the squared prediction error (SPE), and the T^2 statistic was not utilized. In fact, any data-based empirical modeling methods, for example, PCA and other multivariate statistical analysis methods, cannot solve the performance monitoring and fault diagnosis problem as a whole if no other information about process is efficiently utilized. Specifically, the emphasis of most PCA-based research has been placed more on monitoring algorithms using a given set of sensors, and less on actual sensor allocation for efficient detection and identification of process malfunctions. Fortunately, many researchers in other fields have resorted to the design of various sensor networks from a fault diagnosis perspective [Bhushan and Rengaswamy, 2002a; Bhushan and Rengaswamy, 2002b; Kramer et al., 1987; Maurya et al., 2006]. If sensors are suitably located based on the knowledge of fault propagation manner within the process, PCA with optimized sensor locations would certainly enable a great improvement on the process FDI performance.

1.2 R&D Objectives and Original Contributions

The goal of this dissertation was to develop an integrated approach to address sensor placement design in a fault diagnostic architecture with application to a nuclear desalination

system. The contributions of the dissertation are innovative and significant in that, to the best of knowledge, they represent the first comprehensive studies of modeling, simulation, sensor placement design, and fault diagnosis of nuclear desalination processes. The original contributions are presented below.

1. Development of steady-state and dynamic models of a MSF desalination plant using MATLAB and SIMULINK: Steady-state models are applicable for design purposes as well as for parametric studies of existing plants. Dynamic models are suitable for simulating transient behavior, studying control strategies, investigating stability problems, and identifying process interactions.

2. Development of a coupled IRIS-MSF nuclear desalination plant model in SIMULINK: This provides a useful platform to facilitate the study of system transients, control behavior, performance monitoring, and fault diagnosis of a nuclear desalination process.

3. Development of an integrated architecture of performance monitoring and fault diagnostics with emphasis on the importance of optimum sensor placement: Causal graph-based algorithms are developed to perform sensor placement design from a fault diagnosis perspective using different design criteria.

4. Development of an optimal and automated sensor allocation procedure for the IRIS system and the MSF desalination plant: An efficient integer linear programming (ILP) embedded greedy search optimization algorithm was developed to solve the formulated sensor allocation optimization problem. This problem of sensor placement design was also solved optimally in this dissertation using a commercial ILP optimization solver, LINGO 8.0, and the optimal solutions are compared with the greedy search algorithm.

5. Application of the sensor placement design strategy to the IRIS-MSF nuclear desalination plant: A PCA-based fault diagnostic approach with optimized sensor selection is able to achieve the satisfactory FDI performance in both single-fault and multi-fault cases. Moreover, fault diagnostic results are examined using reduced sensor sets (non-optimal sensor selection), as opposed to optimal sensor sets. The comparison study indicates that the reduced sensor set is not

sufficient for the purpose of fault isolation, thus justifying the use of the developed optimal sensor allocation strategy.

6. Development of the effective PCA models using the optimal sensor set obtained through sensor allocation design: This provides guidance for choosing measurements necessary for building PCA models, and eliminates the arbitrary decision making in selecting proper measurements that are needed for process performance monitoring and fault diagnosis.

1.3 Organization of the Dissertation

This document is divided into eight chapters. Chapter 2 reviews the state-of-the-art optimum sensor placement techniques, which consist of three basic components: a model, an evaluation module, and an optimization algorithm. The details of a variety of desalination technologies are also presented in this chapter. This is followed by the discussion of an integrated architecture of a fault diagnostic system in Chapter 3. Both steady-state and dynamic models for a MSF desalination system are developed in Chapter 4. Simulation results are provided and analyzed in this chapter. Chapter 5 presents simulation results of a coupled IRIS-MSF nuclear desalination plant consisting of the MSF dynamic model developed in this dissertation and an existing IRIS plant model in the SIMULINK environment. Next, a complete procedure of implementing optimum sensor placement for fault diagnosis is given in Chapter 6. Different criteria are employed to formulate optimization problems, and a greedy search algorithm is adopted to solve the formulated optimization problems. In Chapter 7 and 8, the developed sensor placement design algorithms and PCA-based fault diagnostic approaches with optimized sensor placement are demonstrated through the applications to a MSF desalination process and an IRIS helical coil steam generator (HCSG) system, respectively. Finally, a summary of the dissertation, with concluding remarks and suggestions for future research, are presented in Chapter 9.

Chapter 2

Literature Review

2.1 Optimum Sensor Placement for Process Systems

The performance of an industrial process is strongly dependent on available sensor measurements. Inaccurate measurements resulting from insufficient measurements or improper sensor placement can significantly deteriorate fault observability and process control. Therefore, sensor placement has received considerable attention and has been studied in different areas.

Traditionally, sensors are placed mainly to meet control or monitoring objectives. The sensor placement for process control is to determine the controlled variables and the manipulated variables to achieve the designed control objectives. The selection of controlled variables is mainly concerned with the process requirements. The general guideline is that the controlled variables should include: (1) non self-regulating variables; (2) environment and equipment safety critical variables; (3) process performance critical variables; (4) variables that have strong interactions with other control variables; and (5) variables with favorable static and dynamic characteristics [Bagajewicz, 2001].

Faulds and King [2000] introduced the sensor placement problem for feedback control. Al-Shehabi and Newman [2001] employed root locus principles to choose optimum sensor positions for aero-elastic vehicle feedback control applications. Giraud and Jouvencel [1995] addressed the problem of sensor selection in an automatic task, such as the process of data fusion, a sensing task, or the design of a perceptual system for a mobile robot. Chen and Li [2002] presented an automatic sensor placement technique for robot vision in inspection tasks. One aspect of the NASA Aircraft Morphing program was to determine the optimum number of active control devices (for example, piezoelectric actuators) and their placement in the structure. In this program, Padula and Kincaid [1999] provided a good review of sensor and actuator placement problems. They grouped the sensor and actuator placement research based on different types of applications (non-aerospace placement problems and aerospace placement problems).

Some recent works have pertained to other nontraditional activities, such as target tracking, fault detection, and reliability analysis. Sensor placement for fault diagnosis has been studied at two different levels: component level and system level. Some of the sensor placement problems attempted to position sensors in a component's range. Critical systems of interest are characterized as large scale systems consisting of multiple components. For such systems, a fault may propagate through several components when it occurs. Therefore, it is possible that sensors can be placed at any of the components to detect the fault. With hundreds or thousands of possible locations of sensors in a system, the selection of a crucial and optimum sensor location, sensor types, and number of sensors poses an important problem that needs to be solved at the system level before the detailed spatial distribution in a component can be determined.

Although a large body of research work has emerged, the various approaches vary only in their choices of the three basic components: a model of the system of interest, a sensor selection module, and an optimization algorithm, as illustrated in Figure 2.1. The following discussion of sensor placement is based on these three components.

2.1.1 Modeling of Systems

To date there have been a plethora of different modeling techniques that have been proposed in the literature. These techniques range from the early attempts using fault trees and digraphs, analytical approaches, and knowledge-based systems and neural networks in more recent studies. From a modeling perspective, most of the techniques can be classified as physics-based modeling or empirical modeling. Physics-based models, also known as first principle models, use physics of the system to predict the nominal conditions of the process. Despite its clear benefit of representing the physical relationships among process variables for all the operational conditions, there are a number of drawbacks of using a physics-based model. It is not uncommon that the underlying physical processes are not completely understood, thus simplifying assumptions must be made to facilitate model development. Assumptions made in model development may not be fully applicable to real world systems, which limit the applicability of the models. In addition, these models are often computationally expensive for complex systems.

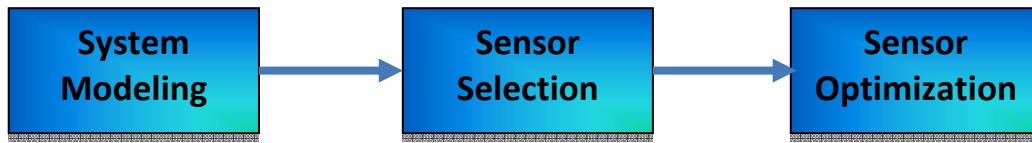


Figure 2.1. Basic components of sensor placement strategy.

Empirical models, on the other hand, use data to fit a model to the relationships seen in real world applications. These types of models are built using process parameter measurements that have been collected over the operational range of the system or process of interest. The relationships among these measurements are used within the model architecture to produce accurate predictions. Empirical models are often preferable to physics-based models because they are simple to develop, they capture real world relationships, and they require limited knowledge of the underlying physical phenomena. In practice, there are two fundamentally different empirical modeling approaches. System identification approach identifies empirical models using data obtained from well-designed experiments. Historical empirical modeling approach develops models from routine operation data saved in a historical database. The empirical models developed from system identification techniques are usually causal models because of the careful control of experiments. On the contrary, historical empirical models are usually not causal models because they can only capture the correlations among the variables contained in the historical data.

Due to the large number of empirical methods available in the literature, an exhaustive survey is not practical. The following discussion, however, provides an overview of the basic concepts of the past research in empirical modeling.

Kalman filter

Kalman filtering was first introduced by Kalman [1960] as a way of designing a state estimator with minimum estimation error. It is well known that the Kalman filter is a recursive algorithm for state estimation and it has found wide applications in aerospace, chemical, and other industrial processes. A good introduction to the general idea of the Kalman filter can be

found in [Maybeck, 1979], while a more complete introductory discussion can be found in [Sorenson, 1970], which also contains some interesting historical narrative. It has been shown that a bank of Kalman filters designed on the basis of all the available possible system models under all possible changes can be used for the isolation purpose. Fathi et al. [1993] included adaptive analytical redundancy models in the diagnostic reasoning loop of knowledge based systems. The modified extended Kalman filter is used in designing local detection filters in their work.

The essential Kalman filter theory can be summarized briefly as follows. Describe a linear finite dimensional stochastic system by a discrete-time state-space model:

$$x(t+1) = Ax(t) + Bu(t) + w(t) \quad (2.1)$$

$$y(t) = Cx(t) + v(t), \quad t \geq 0 \quad (2.2)$$

where $x(t)$ is n -dimensional vector, A , B and C are matrices with suitable dimensions, x_0 has mean $\bar{x}_0$ and covariance Σ_0 ; $w(t)$ and $v(t)$ are Gaussian white noise sequences with zero means and the covariance matrix:

$$E \left\{ \begin{pmatrix} w(t) \\ v(t) \end{pmatrix} \begin{pmatrix} w^T(\tau) & v^T(\tau) \end{pmatrix} \right\} = \begin{pmatrix} Q & S \\ S' & R \end{pmatrix} \delta_{t-\tau} \quad (2.3)$$

where $\delta_{t-\tau}$ is Kronecker's delta, $w(t)$ and $v(t)$ are independent of $\sigma(x_s : s \leq t)$.

In estimating the state $x(t+1)$ based on the observed data $y(t)$ and $u(t)$, the optimal Kalman filter minimizes the function:

$$J = \lim_{t \rightarrow \infty} E(e^T(t)e(t)) \quad (2.4)$$

where $e(t)$ is the estimation error between $x(t)$ and its estimate.

Assume the initial state and noise sequences are jointly Gaussian. Consider the estimator:

$$\hat{x}(t+1) = E \{ x(t+1) | y(t), \dots, y(0), u(t), \dots, u(0) \} \quad (2.5)$$

The filtered state $\hat{x}(t+1)$ satisfies:

$$\hat{x}(t+1) = A\hat{x}(t) + Bu(t) + K(t)[y(t) - C\hat{x}(t)] \quad (2.6)$$

$$\hat{x}_0 = \overline{x}_0$$

The Kalman filter gain $K(t)$ is given by:

$$K(t) = [A\Sigma(t)C^T + S][C\Sigma(t)C^T + R]^{-1} \quad (2.7)$$

where $\Sigma(t)$ is a $n \times n$ state error covariance matrix.

Auto-Associative Kernel Regression

Auto-associative kernel regression (AAKR) is a non-parametric model that uses past normal operational data to correct faulty observations which may be due to system degradation, sensor faults, data acquisition problems, etc [Hines et al., 2008]. The outputs of an AAKR model are the predicted or corrected values of the inputs.

The AAKR model uses a Euclidean distance, which is known as the L^2 -norm, to compare the input query data to the exemplar vectors which make up the model's memory matrix for n inputs.

$$d_i = \sqrt{\sum_{i=1}^n (x_{q,i} - m_i)^2} \quad (2.8)$$

Next, these distances are transformed to similarity measures used to determine weights by evaluating the Gaussian kernel. This kernel is a function of the Euclidean distance, d , and the kernel bandwidth, h .

$$w = K(d, h) = \frac{1}{\sqrt{2\pi}h} e^{-d^2/h^2} \quad (2.9)$$

The final prediction is a weighted sum of the exemplar vectors, m_i .

$$\hat{x}_q = \frac{\sum_{i=1}^{n_m} w_i \cdot m_i}{\sum_{i=1}^{n_m} w_i} \quad (2.10)$$

The parameters to be optimized in an AAKR model are the memory matrix and the kernel bandwidth, h . A researcher must decide how many vectors to include in the memory

matrix and how large to make the bandwidth which indirectly controls how many memory vectors shall be weighted heavily during prediction.

Artificial Neural Networks

Considerable interest has been shown in the literature in the application of artificial neural networks. In general, artificial neural networks can be classified according to the learning strategy such as supervised and unsupervised learning. In supervised learning strategies, by choosing a specific topology for the neural network, the network is parameterized in the sense that the problem at hand is reduced to the estimation of the connection weights. The connection weights are learned by explicitly utilizing the mismatch between the desired and actual values to guide the search. This makes supervised neural networks a good choice for fault classification as the networks are capable of generating, hence classifying, arbitrary regions in space [Cybenko, 1989]. On the other end of the spectrum are neural network architectures which utilize unsupervised estimation techniques. These networks are popularly known as self-organizing neural networks as the structure is adaptively determined based on the input to the network. One such architecture is the ART2 network [Carpenter and Grossberg, 1988].

The most popular supervised learning strategy in neural networks has been the back-propagation algorithm. Most of the work on improvement of performance of standard back-propagation neural network is based on the idea of explicit feature presentation to the ANN. A number of researchers worked on this issue. Fan et al. [1993] discuss the performance gains through the incorporation of functional inputs in addition to the normal inputs to the neural networks. Incorporation of knowledge into the ANN framework for better diagnosis is discussed by Farrell and Roat [1994]. Data processing and filtering is shown to lead to significant performance improvement and reduced training time. Tsai and Chang [1995] propose the integration of feed forward ANN with recurrent ANN for better performance.

There are also other architectures such as self-organizing maps [Kohonen, 1984]. The objective of these methods is to give credit for patterns that are similar to group together. The similarity measure is usually a distance measure. Whenever a pattern is seen that is not similar (in a distance metric sense) to any of the previously formed classes, a new class is formed and

the pattern is retained as the defining pattern for the new class and similarity is measured with respect to this pattern. The crucial elements in these kinds of architectures are the distance metric that one chooses and the threshold for similarity. Clustering is a technique to group samples so as to maximize the separability between these groups. Clustering algorithms specify the number of groups and maximize an objective function that is a measure of separability of these groups. In this manner, clustering becomes a well-defined optimization problem. In the clustering process credit is given to patterns exhibiting similar characteristics. Clustering procedures need two important components. First, they need a measure for estimating similarity between different data points. Without this no credit can be assigned for patterns that are similar. Second, one needs representative patterns against which the similarity of other patterns can be checked.

The most popular clustering algorithm proposed in the literature is the K-means clustering algorithm [Duda and Hart, 1973]. K-means clustering pre-supposes the number of clusters needed and would cluster the data accordingly. It utilizes all the cluster centers so that each of the clusters is guaranteed at least one pattern. Kohonen's self-organizing maps [1984] identify the cluster center closest to the training pattern and update this cluster center and all its topological neighbors. K-means clustering can be shown to be a special case of Kohonen's clustering algorithm. In Kohonen's algorithm, after the neighborhood is decided, the algorithm makes all the clusters in the neighborhood to be the winners of the pattern. This leads to the problem of gravity where all the cluster centers migrate towards dense regions leaving less dense regions unrepresented. Self-organizing neural network structures such as the ART2 network [Carpenter and Grossberg, 1988] have also been extensively used in fault diagnosis. Whiteley and Davis [1994] demonstrate the use of ART2 network for the interpretation of sensor data. Chen et al. [1999] and Wang et al. [1999] discuss the integration of wavelets with ART networks for the development of diagnostic systems. For a collection of papers on the application of neural networks in solving engineering problems, interested readers are referred to Venkatasubramanian and McAvoy [1992] and Bulsari [1995].

Ordinary Least Squares Regression

Ordinary least squares regression is a basic algorithm for illustrating inferential techniques. It is also useful for performance bench marketing. Its derivation can be found in [Hastie, 2001] and it is restated here.

Given a linear model:

$$y = X\beta + \varepsilon \quad (2.11)$$

where y is a measurement with N observations; X is p vector of inputs, each vector has N observations; $\varepsilon \sim N(0, \sigma^2)$ is the noise.

Estimate the coefficients $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)^T$ by minimizing the residual sum of squares:

$$RSS(\hat{\beta}) = \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \sum_{i=1}^N (y_i - \hat{\beta}_0 - \sum_{j=1}^p x_{ij} \hat{\beta}_j)^2 \quad (2.12)$$

$$RSS(\hat{\beta}) = (y - X\hat{\beta})^T (y - X\hat{\beta}) \quad (2.13)$$

The solution is derived by setting derivative of the cost function to zero.

$$\frac{\partial RSS(\hat{\beta})}{\partial \hat{\beta}} = -2X^T (y - X\hat{\beta}) \quad (2.14)$$

$$\frac{\partial^2 RSS(\hat{\beta})}{\partial \hat{\beta} \partial \hat{\beta}^T} = -2X^T X \quad (2.15)$$

Assuming that X is nonsingular and hence $X^T X$ is positive definite, set the first derivative to zero to minimize the RSS.

$$X^T (y - X\hat{\beta}) = 0 \quad (2.16)$$

The unique solution can be obtained as:

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (2.17)$$

The predicted values at the training inputs are:

$$\hat{y} = X\hat{\beta} = X(X^T X)^{-1} X^T y \quad (2.18)$$

Parameter uncertainties can be estimated using an estimate of the noise ε .

$$\text{Var}(\hat{\beta}) = (X^T X)^{-1} \sigma^2 \quad (2.19)$$

$$\hat{\sigma}^2 = \frac{1}{N-p-1} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.20)$$

Assume the linear model is correct and the deviations of Y around its expectation are additive and Gaussian.

$$Y = \beta_0 + \sum_{j=1}^p X_j \beta_j + \varepsilon \quad (2.21)$$

$$\varepsilon \sim N(0, \sigma^2) \quad (2.22)$$

Then,

$$\hat{\beta} \sim N(\beta, (X^T X)^{-1} \sigma^2) \quad (2.23)$$

$$(N-p-1)\hat{\sigma}^2 \sim \sigma^2 \chi_{N-p-1}^2 \quad (2.24)$$

$\hat{\beta}$ and $\hat{\sigma}$ are statistically independent. The mean squared error of an estimator $\hat{y}$ in estimating y is:

$$\text{MSE}(\hat{y}) = E(\hat{y} - y)^2 = \text{Var}(\hat{y}) + (E(\hat{y}) - y)^2 \quad (2.25)$$

Least squares estimates have the smallest variance among all linear unbiased estimates. However, there may be a biased estimator with smaller mean squared error. Such an estimator would trade a little bias for a larger reduction in variance. Biased estimators, such as ridge regression, are commonly used. Any method that shrinks or sets to zero some of the least squares coefficients may result in a biased estimate.

2.1.2 Sensor Selection Evaluation

The goal of the sensor selection process is to provide a suite of sensors that fulfill specified performance requirements within a set of system constraints. These performance requirements are defined as the figures of merit (FOM) of the system, and considerable research has focused on how to represent these FOM in algorithmic form. The following list of general FOM categories was selected after reviewing the available literature and research.

Observability

Of ultimate importance for any sensor suite is the concept of observability. Basically, observability is the capacity of the sensor network to provide information about the state parameters deemed important for performance monitoring, health assessment, and/or control of the system. This information may be provided as direct measurements of the system parameters or, as in data reconciliation [Zhao, 2005b], the reconstruction of unobservable system parameters based upon observable or sensed variables. Approaches vary in the determination of the observability of the sensor network. Some treat observability as a true or false property, while others devised more sophisticated schemes to display the degree of observability.

Some approaches used graph-based analysis for which the structural information of a system is represented by a directed graph (DG), called a digraph. Observability is then based on analysis of cut-sets and cycles generated from the digraph. Kretsovalis [1987] proposed two graphically-based algorithms for classification of observable and redundant variables. Luong et al. [1994] also used graph-based analysis to establish an incidence matrix, relating the process relationships to the state variables qualitatively. Decomposition of this incidence matrix using a Gauss-Jordan elimination process identified whether an unmeasured variable was observable and whether a measured variable was redundant. Using graph theory and cut-sets, Bagajewicz and Sanchez [1999] defined the degree of observability and degree of redundancy for a sensor network. For an unmeasured variable, the degree of observability is the maximum number of sensors that can be eliminated with the variable still observable; for measured variables, the degree of redundancy is the maximum number of sensors that can be eliminated and the measurement remains redundant. These concepts were combined to define a degree to which system variables can be estimated by the sensor network.

Other approaches attempted to define the degree of observability by analyzing the numerical relationships of the system process. By taking advantage of linear state space system theory, observability and controllability can be represented in matrix form. The system is observable or controllable if the observability or controllability matrices are of full rank. These represent binary conditions as to whether the system is observable/controllable or not. Dochain et

al. [1997] considered both the rank and condition number of the observability matrix for an optimal sensor placement in a bioreactor system. The Gramian of both the observability and controllability matrices have been used by researchers [Muller, 1972; Van den Berg et al., 2000] to develop scalar metrics that indicated the degree of each respective property.

A large amount of research has been devoted to control design and actuator/sensor placement locations, particularly in structural-type problems. Papadopoulos and Garcia [1998] investigated a number of quantitative techniques used to optimize structural sensor placement to select sensor locations that would observe the maximum amount of system information, which is energy in this case. Each technique attempted to characterize the amount of energy the sensors would observe and eliminate potential sensor locations that had minimal observability. Hac and Liu [1993] proposed a methodology that incorporated eigenvalues from both the observability and controllability matrices to select sensor locations for the control of flexible structures.

Several researchers used the predictive error of a given sensor suite relative to a specific reconciliation technique as the performance metric. After performing matrix decomposition on the linear system model coefficients, Madron [1992] proposed a simple local optimization of this baseline set of sensors by computing the prediction precision of unmeasured required state parameters. The baseline sensor suite performance was compared to suite variations where a single sensor was removed and the objective function recomputed. Chmielewski et al. [2002] developed the framework for the justification and incorporation of the error covariance matrices for sensor suite performance properties and demonstrated how that could be optimized with other suite performance constraints using mixed linear integer programming techniques. Musulin et al. [2005] proposed a sensor selection process that maximized a Kalman Filter performance, via the error covariance matrix diagonal terms. He pointed out that although the objective is to maximize the performance of the filter for all variables, maximizing the performance for one particular variable may not be compatible with maximizing it for others, resulting in a conflict. Therefore they proposed two possible approaches; one method would select the variable with the lowest performance value. The second approach would evaluate the performance of each variable relative to an “ideal” sensor suite performance, combining into an overall performance metric. Mushini and Simon [2005] used a similar approach of maximizing a Kalman Filter

performance for an aircraft gas turbine engine. The metric was defined as the summation of the error covariance matrix diagonal terms normalized to the reference or ideal system performance.

Sensor Reliability

Regardless of the purpose of the sensor network, consideration must be given to the availability of the sensor signal information when it is required. Sensor faults can be common and the potential for interruption of information flow and how that is to be handled is an important consideration in the sensor network design analysis.

Luong et al. [1994] investigated sensor network reliability from a control perspective that the reliability of an instrumentation system is the probability that information required for control are available through measurements or deduction during some time period. Based on that idea, an overall sensor network reliability metric was defined as an expression of the individual sensor reliabilities. Sensor reliabilities were expressed as a function of time; therefore, the integration of the time profile for each sensor network was used as a quantitative comparison metric between competing networks.

Ali and Narasimhan [1993] introduced the concept of reliability of the estimation of variables, which relates the sensor reliability with its availability to provide system state estimations. The focus of the research was for a given sensor network, how many different ways can a variable be estimated and if sensors fail, can a variable still be estimated. The network objective function was defined as the minimum product of the input sensor reliabilities for each estimated parameter. The initial processing routine involved graph theory concepts of chords, cut-sets, and spanning trees to establish sets of estimated variables and their associated sensors used to observe the variable. This objective function was extended to include hardware redundancy (measuring a variable using more than one sensor) and spatial redundancy (more than one way of estimating a variable) [Ali and Narasimhan, 1995]. Within the graph theory algorithm, the sensor network can be optimized with respect to only a single criterion; therefore, this objective function was adapted for processing with genetic algorithms [Sen et al., 1998] and mixed integer nonlinear programming (MINLP) [Bagajewicz and Sanchez, 2000] which enabled it to be evaluated with other performance metrics, such as sensor costs.

Another consideration of the sensor network is the performance in the presence of sensor faults. Sensor networks must be robust to sensor faults, both in being able to perform adequately with a faulty sensor in the network and being able to continue performance adequately if a sensor is removed. Bagajewicz [1997] proposed a set of performance metrics for a sensor suite that defined the ability of the sensor network to detect sensor faults (error detectability) and handle sensor faults relative to reconciliation precision performance (availability - precision after failed sensor removed, and resilience - precision in the presence of a sensor fault). Bagajewicz and Cabrera [2002] extended this research from the graphical tree-search-solution procedure to the explicit MINLP formulation and also extended the theory for explicit consideration of hardware redundancy.

Fault Detectability/Fault Discriminability

For fault diagnosis, the ability of the sensor network to detect and discriminate failure modes and anomalous conditions is of prime importance. This is an extension of the observability analysis where specific state variables or groups of state variables and their response are indicative of health conditions. The performance metric must define the sensor network's ability to distinguish these fault conditions from nominal operation (fault detection) and distinguish these fault conditions from one another (fault discrimination).

Significant research is focused on the qualitative or heuristic representation of the fault progression process. Research conducted by Raghuraj et al. [1999], Bhushan and Rengaswamy [2000a; 2000b], and Bagajewicz et al. [2004] utilized directed or sign directed graphs to generate fault signatures. A directed graph of a hypothetical process with faults F_i , connecting to sensors S_i , indicating that the fault will affect the reading of the corresponding sensor, is displayed in Figure 2.2. These signatures provided only an indication that the sensor should respond to the particular fault condition, but the actual signature, magnitude or rate, was not utilized. Spanache et al. [2004] used the physical constraints of the system to develop a set of analytical redundancy relationships between the measured and unmeasured variables and qualitatively established an influence matrix between system components and possible sensor parameters. Fault signatures

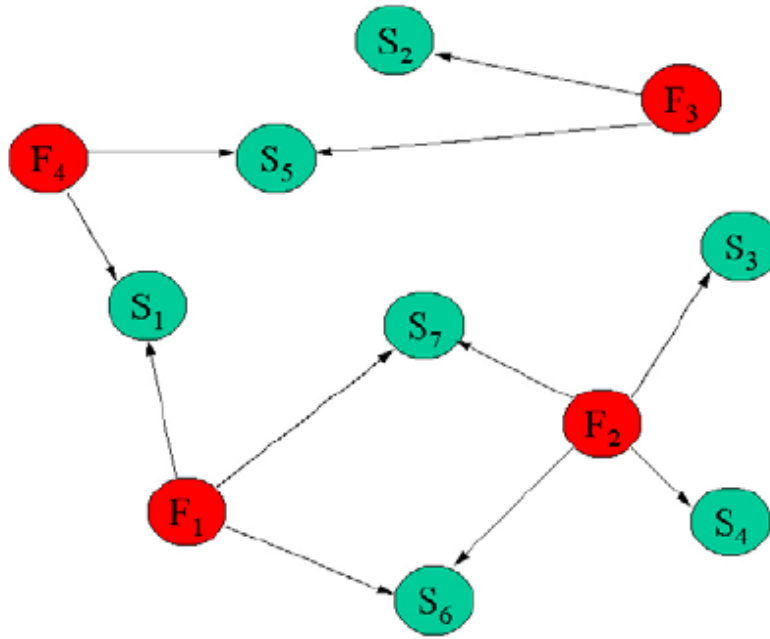


Figure 2.2. Directed graph of a hypothetical system.

were assigned to a failed component without regard to specific component failure modes, and hence a fault signature matrix was generated between the component faults and the influenced analytical redundancy relationships. Narasimhan et al. [1998] used temporal causal graphs, and Yan [2004] used a CAD system model to develop qualitative fault signatures. Representing the fault signature response qualitatively avoids the cost of simulating the fault in hardware and/or software and often this is the only level of fault information available during the early development of a system.

In an effort to incorporate more fault information, Zhang [2005] proposed a quantified directed graph (QDG) to model the fault propagation. For the QDG, each node represented a potential sensor with an associated signal-to-noise ratio (SNR), and each arc contained the fault propagation gain and direction between the sensors, along with fault propagation time. A sensor detectability value was computed for each sensor in the suite for a predefined set of faults. Fault detectability was averaged over the available sensor suite and evaluated against an assigned lower bound. Fault resolution was still treated somewhat qualitatively by comparing sets of sensors influenced by each fault.

Azam et al. [2004] proposed a method that used a system model and reliability data to establish cause/effect dependencies between the faults of interest and the effects of faults on observable system parameters. The detection and false alarm probabilities associated with the failure source and observable discrepancy were utilized. A general multiple fault diagnosis algorithm was used to search for the most likely candidate fault that explained the set of observed discrepancies. Three diagnostic performance metrics were established for each potential sensor based on the decision probability error from a number of simulation runs. Selection evaluation was performed for candidate sensor networks by summing these performance metrics over the sensor suite and incorporating cost constraints.

Santi et al. [2005] utilized a high-fidelity dynamic model of a rocket propulsion system to generate a simplified linear model. Health parameters were identified in the linear model that represented specific fault conditions. This research focused on mean-shift faults in the system. An inverse diagnostic model was developed which predicted the change in health parameters based on sensor inputs. Fault trajectories were generated and metrics were established for detection timeliness and fault discrimination. Fault detection required sufficient measurement deviation to discriminate a fault from normal operation variance. Detection thresholds were established based on the historical information of measurement variance, anticipated process variance, and defined false alarm constraints of the system. The diagnostic performance metrics were based on the distance measured between fault trajectories for fault discrimination and inferred time to detection threshold areas for detection timeliness. In addition, fault risk reduction measures were incorporated into the evaluation process to allow weighting of individual failure modes based on criticality and occurrence rates.

2.1.3 Sensor Optimization Techniques

After a FOM is selected, an algorithm needs to be selected to optimize this FOM. Various optimization algorithms, from random search to heuristic algorithms such as genetic algorithms (GA), have been used for optimizing sensor allocation. Random search is suitable for a small and simple sensor placement problem since it is straightforward and easily implemented. But it is time consuming and inefficient when dealing with a large system. A simulated annealing (SA)

method [Johnson et al., 1989] based on random search is used to select a single random subset while seeking to improve the cost function by moving to one of the nearest neighbors of the selected subset. The Tabu search method [Glover and Laguna, 1997] uses probabilistic events for the search of better solutions. Genetic algorithms, based on the Darwinian principle of natural selection, are widely applied in different domains, among the heuristic methods. Sen et al. [1998] designed a sensor network for the linear mass flow process using the GA method. Other examples using the GA method to solve sensor placement problems can be found in [Furuya and Haftka, 1996] and [Ponslet et al., 1993].

Other optimization approaches include graph-based methods. A greedy search algorithm based on a bipartite graph was proposed by Raghuraj et al. [1999] to generate a minimal set of sensors to meet observability and maximum resolution requirements. A bipartite graph is one whose vertex set can be partitioned into two sets in such a way that each edge joins a vertex of the first set to a vertex of the second set. Whenever there is a directed path from a root node to a key component, an arc from that root node to the key component is drawn in the bipartite graph.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Eberhart and Kennedy [1995], inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as genetic algorithms. The system is initialized with a population of random solutions and searches for optima by updating the generations. However, unlike the GA method, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. The fitness value is also stored. This value is called “*pbest*”. Another best value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighborhood of the particle. This location is called “*lbest*”. When a particle takes all the population members as its topological neighbors, the best value is a global best and is called “*gbest*” [Zhang, 2004], as shown in Figure 2.3.

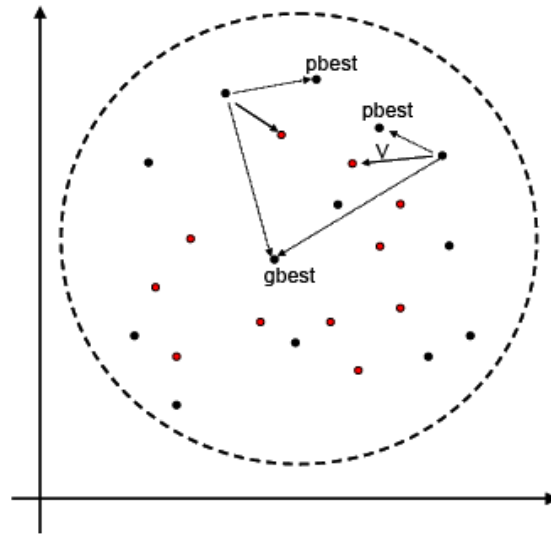


Figure 2.3. Particle swarm optimization.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its “*pbest*” and “*lbest*” locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward “*pbest*” and “*lbest*” locations. In the past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper manner compared with other methods.

Another reason that PSO is attractive is that there are few parameters to adjust. One version, with slight variations, works well in a wide variety of applications. Particle swarm optimization has been used for approaches that are employed across a wide range of applications.

2.2 Seawater Desalination Methods

This section introduces the classifications of desalination methods. Also, the principles and operational variables of several widely used desalination technologies are described in this section.

2.2.1 Classification of Desalination Processes

Desalination is a process of separating dissolved salts from saline water. Many desalination technologies have been developed based on different principles of separation. Some of them have been successfully developed and were discussed in detail in [IAEA, 2002], but only a few of them reached commercial operation. Figure 2.4 illustrates the major desalination processes.

Desalination processes can be broadly categorized into two main types: processes using heat and process using electricity. The first type of processes is of mainly the distillation processes, multi-stage flash (MSF) or multi-effect distillation (MED). Vapor compression (VC) is a distillation process but it uses electricity, just as the membrane based processes like the reverse osmosis (RO) and the electro-dialysis (ED). Of these, the most commonly used processes are MSF, MED and RO. VC is often combined with MED.

The applicability of any process depends on the salt concentration in the feed water and on its water unit cost. Distillation is the oldest and most commonly-used desalination techniques. In this process, evaporation of the saline water and condensation of the generated vapor occur to obtain fresh water. This process produces fresh water with a better quality as compared with crystallization and membrane processes.

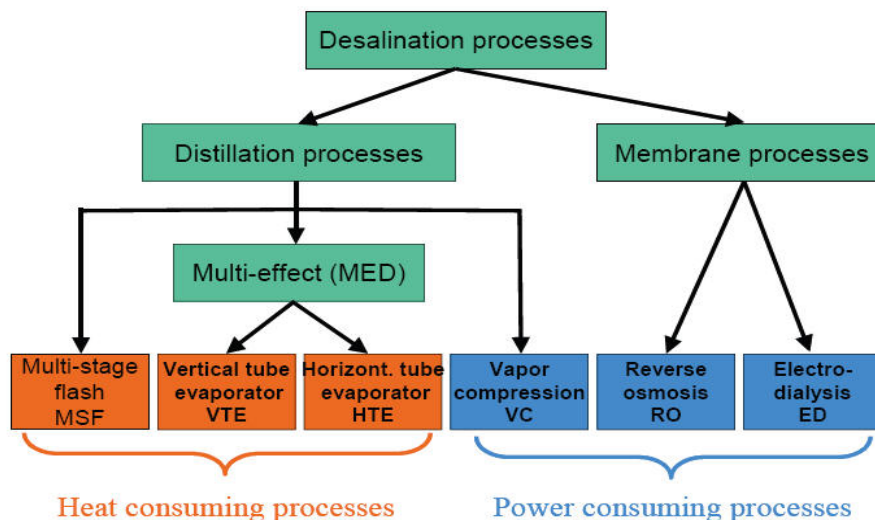


Figure 2.4. Classification of desalination processes [IAEA, 2007].

2.2.2 Distillation Desalination Systems

In distillation processes (MSF or MED), seawater is heated to evaporate into pure vapor that is subsequently condensed. The heat energy required for distillation is usually supplied as low pressure saturated steam, which may be extracted from the exhaust of a low pressure turbine, from a crossover steam duct or from a dedicated, heat only, plant.

Vapor Compression

Vapor compression distillation uses mechanical energy rather than thermal energy. It is based on a simple principle. Saline water is sprayed over an evaporator tube bundle. The vapor formed at some temperature and pressure is then compressed either thermally in a steam ejector, or mechanically (high and low pressure) in a compressor, causing the condensation temperature and pressure to increase and the volume to decrease. Compressed vapor is passed through the evaporator bundle, where it condenses and forms distilled water. The heat of condensation could be recycled to evaporate more brine. Most vapor-compression plants have single effects, but a multi-effect configuration could be used for a larger product capacity. Figure 2.5 illustrates the principle of vapor compression. The vapor-compression process consumes a small amount of energy and has a low operating cost. However, its capacity is limited, and the quality of water produced and maintenance costs do not match those of other distillation processes.

Multi-Effect Distillation

Figure 2.6 shows the schematic flow diagram of a MED process, using horizontal tube evaporators (HTE). In each effect, heat is transferred from the condensing water vapor on one side of the tube bundles to the evaporating brine on the other side of the tubes.

This process is repeated successively in each of the effects at progressively lower pressure and temperature, driven by the water vapor from the preceding effect. In the last effect, at the lowest pressure and temperature, the water vapor condenses in the heat reject heat exchanger, which is cooled by incoming seawater. The condensate distillate is collected from each effect.

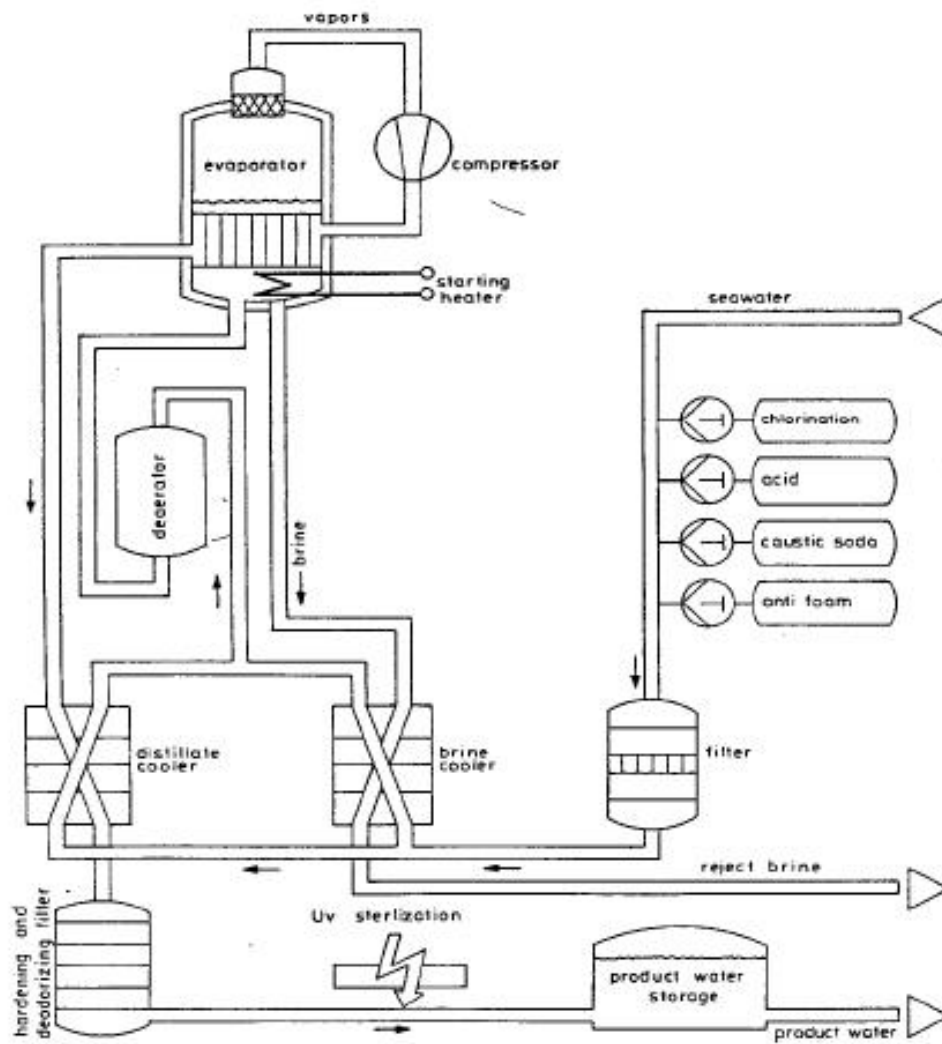


Figure 2.5. Principle of vapor compression [Khan, 1986].

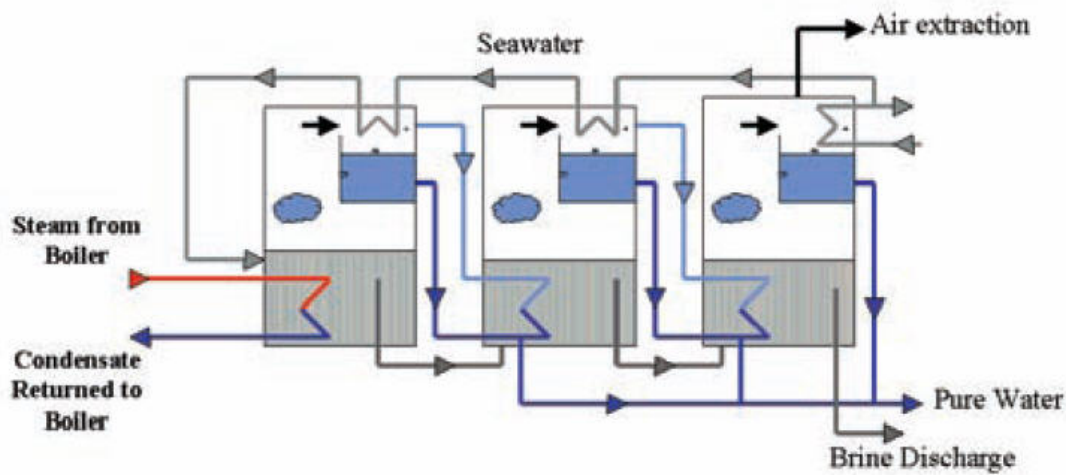


Figure 2.6. Schematic diagram of a MED system [IAEA, 2007].

According to the direction of vapor and brine flow, there are “forward feed” and “backward feed” arrangements. In forward feed MED plants, vapor and brine move through the evaporators as parallel flows from the first high pressure evaporator to the last low pressure one. The pre-heating of feed water occurs in separate heat exchangers. In backward feed MED plants, vapor and brine move through the evaporators in opposite directions, whereby feed water pre-heating is eliminated.

Currently, MED processes with the highest technical and economic potential are the low temperature horizontal tube multi-effect process (LT-HTME) and the high temperature vertical tube evaporation process (HT-VTE). The main differences between LT-HTME plants and HT-VTE plants are in the arrangement of the evaporation tubes, the side of the tube where the evaporation takes place and the evaporation tube materials used. In LT-HTME plants, evaporating tubes are arranged horizontally and evaporation occurs by spraying the brine over the outside of the horizontal tubes creating a thin film from which steam evaporates. In HT-VTE plants, evaporation takes place inside vertical tubes. Furthermore, in LT-HTME plants, the maximum brine temperature is limited to 70 °C, in order to avoid corrosion and scaling problems. Most LT-HTME plants now use low cost materials such as aluminum for heat exchanger and carbon steel as shell material.

Multi-Stage Flash Distillation

Figure 2.7 shows the operating principle of a MSF plant. Consider a vessel under vacuum, isolated from its environment and containing only seawater in equilibrium with its vapor at temperature t and pressure p . When a heating fluid (generally hot water) at a temperature $t + \Delta t$ is introduced in the vessel, and if the pressure p is less than the saturation pressure, an instantaneous vaporization will be produced by a flash. The latent heat of vapor can then be transferred to cold seawater tubes passing through the vessel and the vapor condensed and recovered in the receptacle.

Figure 2.8 illustrates the schematic flow diagram of a MSF system. Seawater feed passes through tubes in each evaporation stage where it is progressively heated. Final seawater heating occurs in the brine heater by the heat source. Subsequently, the heated brine flows through nozzles into the first stage, which is maintained at a pressure slightly lower than the saturation pressure of the incoming water. As a result, a small fraction of the brine flashes forming pure steam.

The heat needed to flash the vapor comes from cooling of the remaining brine flow, which lowers the brine temperature. Subsequently, the produced vapor passes through a mesh demister in the upper chamber of the evaporation stage where it condenses on the outside of the condensing brine tubes and is collected in a distillate tray. The heat transferred by the condensation warms the incoming seawater feed as it passes through that stage. The remaining brine passes successively through all the stages at progressively lower pressures, where the process is repeated. The hot distillate flows as well from stage to stage and cools itself by flashing a portion into steam which is re-condensed on the outside of the tube bundles.

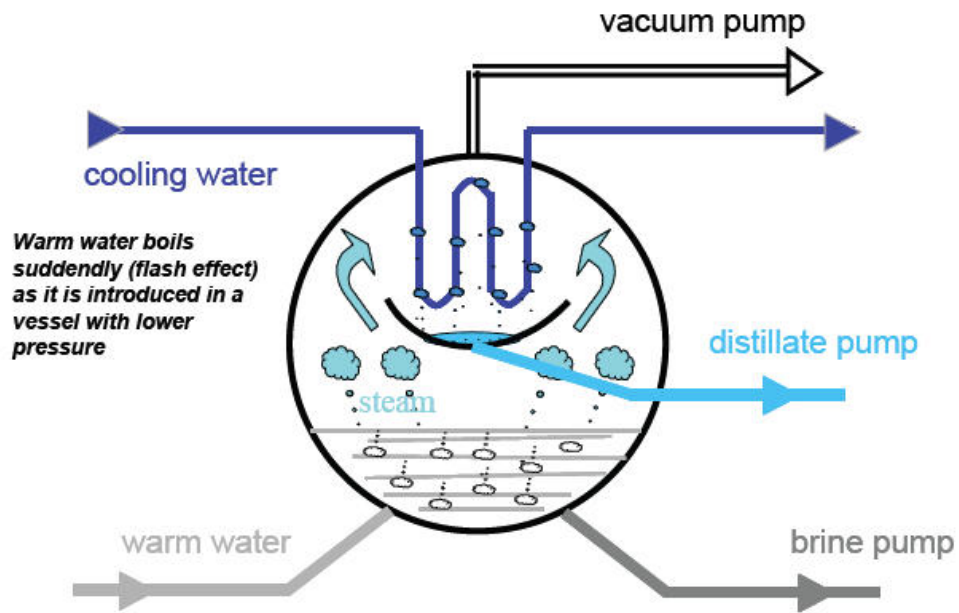


Figure 2.7. Operating principle of a single stage MSF [IAEA, 2007].

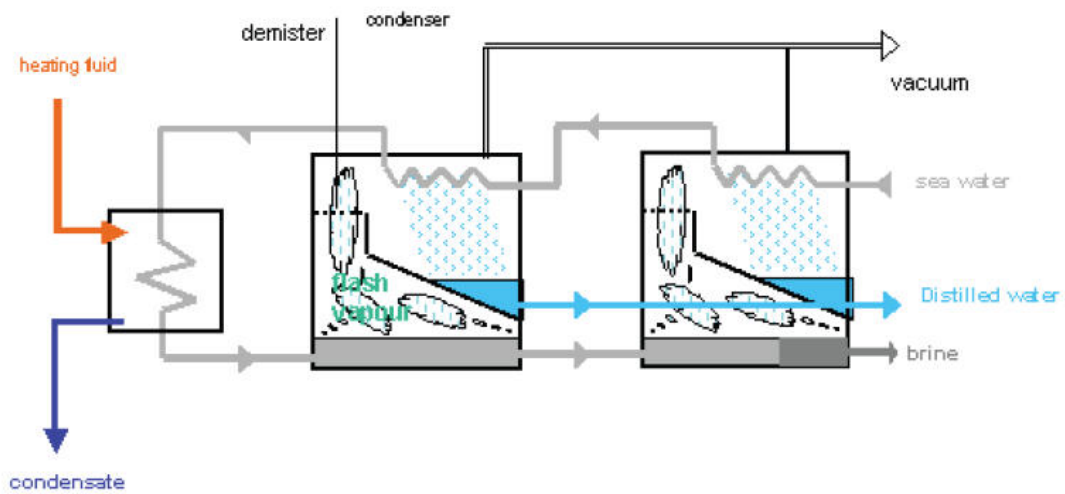


Figure 2.8. Schematic diagram of an industrial MSF design [IAEA, 2007].

MSF plants need pre-treatment of the seawater to avoid scaling by adding acid or advanced scale inhibiting chemicals. If low cost materials are used for construction of the evaporators, a separate deaerator is to be installed. The vent gases from the deaeration together with any non-condensable gases released during the flashing process are discharged to the atmosphere. There are two principal arrangements in MSF systems: the brine recycle mode (MSF-BR), and the once-through mode (MSF-OT). The majority of the MSF plants that are built use the brine recycle mode. The brine re-cycle mode was invented in the early years of desalination when seawater corrosion materials and advanced additives were not available or too expensive. Today, corrosion resistant materials are available at reasonable costs as well as high temperature, cost effective anti-scalants. Therefore, MSF-OT plants have already been successfully applied.

2.2.3 Membrane Desalination Systems

Reverse osmosis and electro dialysis are the two most important membrane processes. To affect salt separation, RO uses hydraulic pressure, whereas ED uses electric current.

Reverse Osmosis

Osmosis is a natural process in which water molecules migrate across a semi-permeable membrane from a solution of low concentration (e.g. pure water) into a solution of higher concentration (e.g. seawater). Reverse osmosis is a separation process in which pure water is “forced” out of a concentrated saline solution by flowing through a membrane at a high static trans-membrane pressure difference (Figure 2.9). This pressure difference must be higher than the osmotic pressure between the solution and the pure water.

The saline feed is pumped into a closed vessel where it is pressurized against the membrane. As a portion of water passes through the membrane, the salt content in the remaining brine increases. At the same time, a portion of this brine is discharged without passing through the membrane.

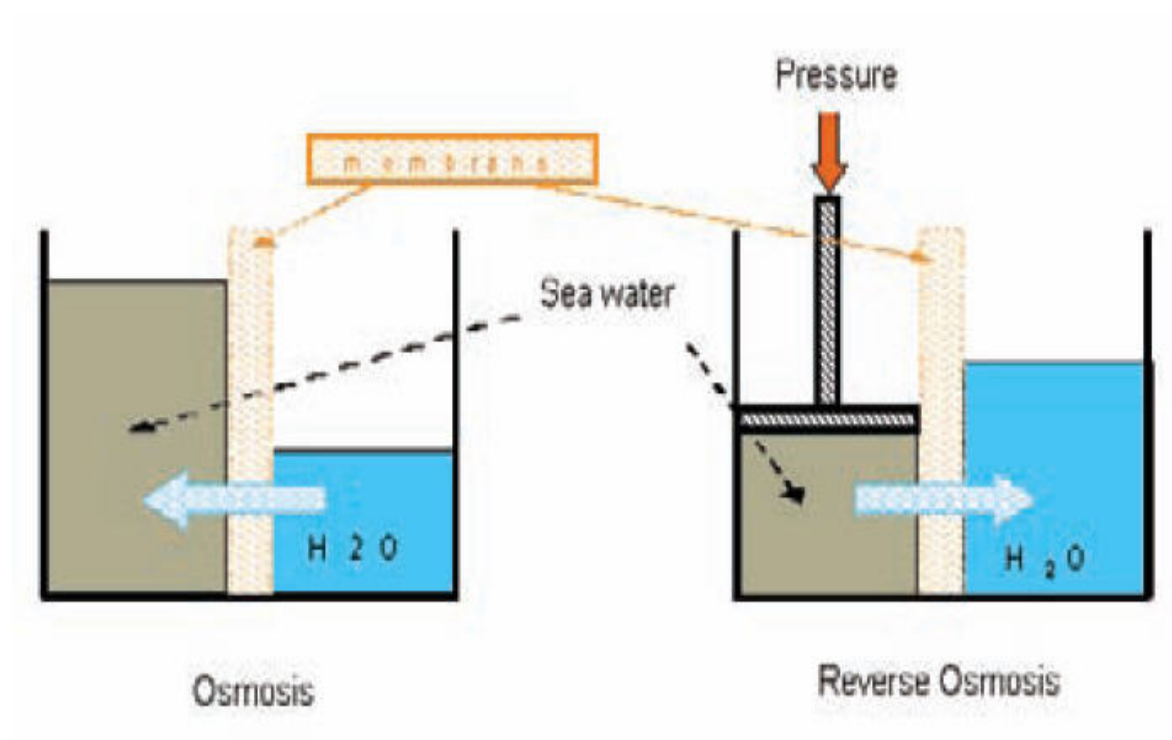


Figure 2.9. Osmosis and reverse osmosis processes [IAEA, 2007].

Electro-Dialysis

Figure 2.10 illustrates an ED process, where two types of membranes are used. The cation membrane allows only cations (positive ions) to permeate, and the anion membrane allows only anions (negative ions) to permeate. These exchange membranes are alternately immersed in salty water in parallel, and an electric current is passed through the liquid. The cations will migrate to the cathode, and the anions will migrate to the anode. Therefore, water passing between membranes is split into two streams. One is pure water, and the other is concentrated brine. Because ED uses energy at a rate directly proportional to the quantity of salts to be removed, this process is more useful in desalting brackish water.

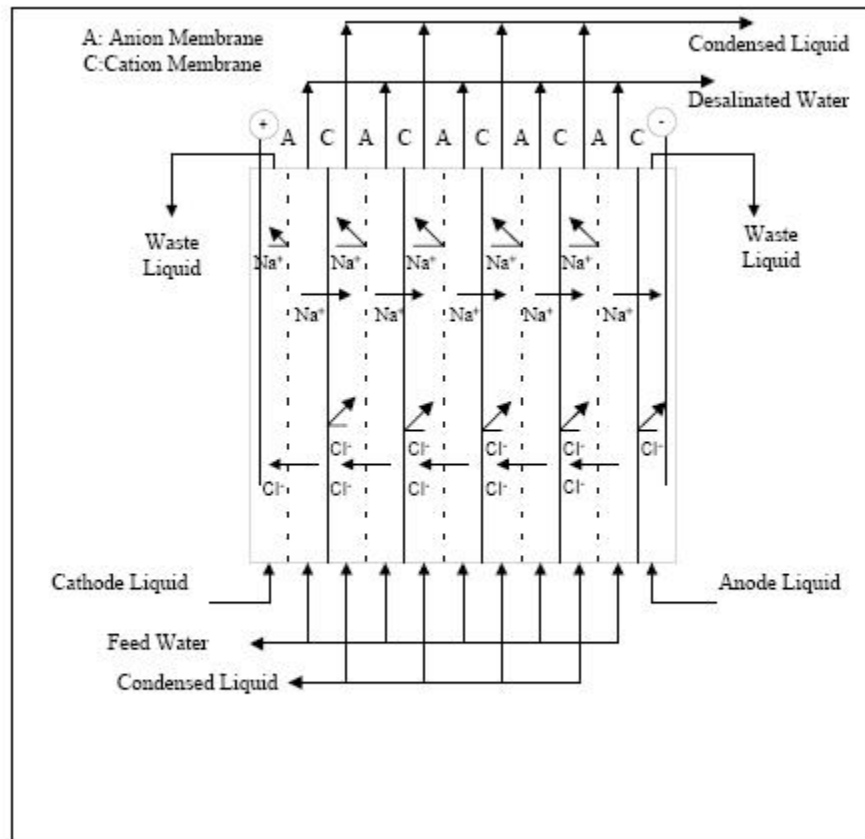


Figure 2.10. Principle of electro dialysis [Khan, 1986].

Chapter 3

Integrated Architecture of a Fault Diagnostic System

Large industrial systems, such as power plants, chemical plants, pulp and paper plants, metals and other manufacturing systems, usually employ a multitude of sensors to monitor and control the operating state, which also makes it possible to improve the operational safety and to perform condition-based maintenance (CBM). This results in significant reduction in plant downtime and considerable amount of maintenance cost savings.

Sensors are crucial for automation and process control, both for achieving performance and for guaranteeing plant safety. The technological drive towards automated plants and high energy-intensive processes demands greater intelligence, availability, and reliability of instrumentation. Even when a process is intrinsically safe, loss of a sensor would lead to loss of process control and hence a sustained period of poor quality production. A sensor may be used in control systems, safety systems, or for condition monitoring. For condition monitoring, sensor output is normally displayed on a panel or on a computer screen, and not used as an input parameter of any system control. If the faulty sensor is the one used as a monitoring device, then the effect of its fault is one-dimensional, that is, the fault will be observed in its own signature and nowhere else. A multi-dimensional fault shows its effects in multiple measurements. The faults with control-related sensors are multi-dimensional because the fault effects can be propagated through control loops and subsequently disturb many other variables throughout the entire process. For instance, as soon as the sensor drifts, the controller notices the sensor signal variations and actuates in the opposite direction, in an attempt to correctly match the set point.

The detection and isolation of faults in a large industrial system, such as a nuclear power plant or a fossil power plant, is generally complicated due to measurement limitations and uncertainties, various types of processes and equipment, and interactions among the sub-systems. A fault diagnostic system is expected to be able to track incipient equipment and sensor degradation, and to assist in enhanced operation and maintenance planning. One approach to characterizing the measurements is to build empirical models, and to use them in conjunction

with a set of rules that defines the cause-and-effect relationships among the process variables. The fault diagnostic module developed in this work is a two-step approach: (1) Development of empirical models for fault residual generation; (2) Detection and isolation of incipient faults based on these residuals. The residuals are computed as the difference between the measured process variables and the empirical model's estimations for those process variables. It is assumed that the residuals are small during normal plant operations. When a fault occurs in a device, equipment, or the process itself, the model residuals deviate from normal allowable values, and indicate a possible abnormal situation.

Sensor placement design is a critical component of a fault diagnostic system, and therefore it is emphasized in the developed integrated approach as illustrated in Figure 3.1. Whenever a process encounters a fault, the effect of the fault is propagated to some or all the process variables. The main objective of fault diagnosis is to observe these fault symptoms and determine the root causes of observed behavior. The ability of the sensor network to detect and discriminate failure modes and anomalous conditions is crucial for the efficiency of the fault diagnostic system. With hundreds of process variables available for measurement in the nuclear plant and the desalination process, selection of optimum number of sensors and their locations poses a unique challenge, and it is indeed an issue that must be addressed in the design phase.

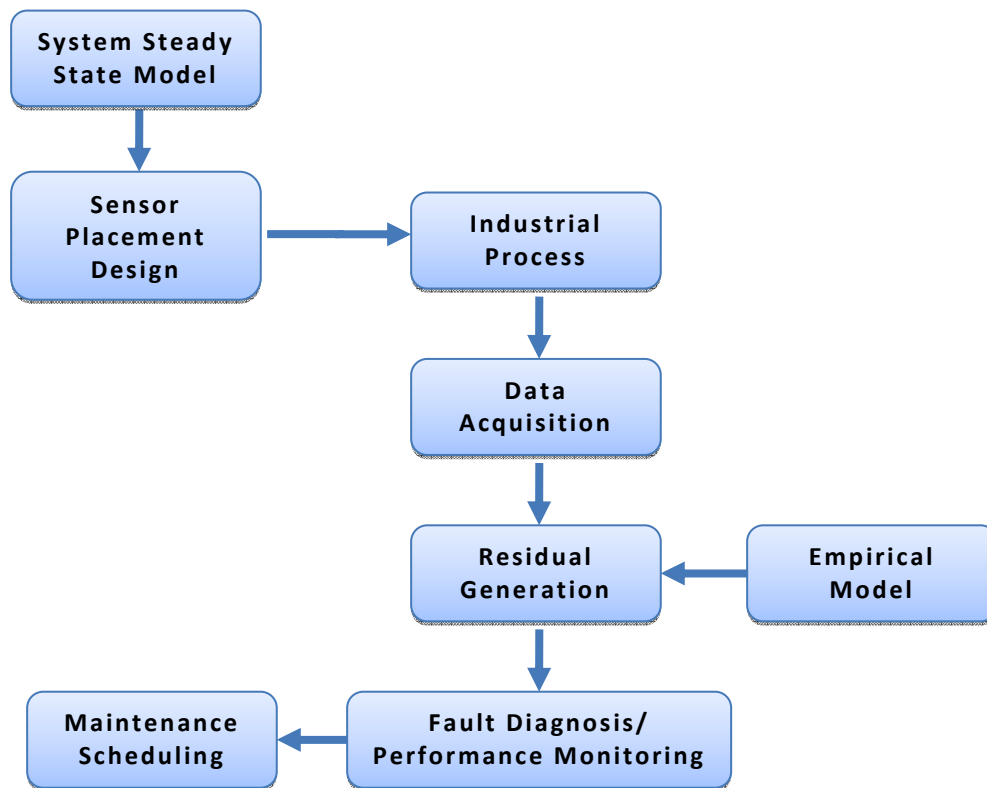


Figure 3.1. An integrated architecture of a fault diagnostic system.

Chapter 4

Modeling and Simulation of a Multi-Stage Flash Desalination System

4.1 Introduction

Among a variety of desalination technologies, multi-stage flash (MSF) contributes substantially to the desalinating capacity in the world. Market share of the MSF processes accounts for 70% of all seawater desalination processes [El-Dessouky et al., 1999]. Thus, issues such as optimization of the operation and minimization of the corresponding environmental impact are of the greatest importance. With the objective of addressing all these aspects, mathematical models prove to be a very useful tool. Steady-state models help in plant design and in fixing operating conditions, whereas dynamic models are required for studying control system design, performance optimization, and fault diagnosis and prognosis. The physics model of a MSF desalination plant gives rise to a nonlinear boundary value problem. The stage-by-stage approach to solve the model is characterized by slow convergence and stability problems. Gluek and Bradshaw [1970] were among the earliest to develop a model of MSF plants with a high degree of rigor and few qualifying assumptions. The model takes into account the variation of heat transfer coefficient, vaporization from the product tray, etc. Helal et al. [1986] reported a tri-diagonal matrix (TDM) model for steady-state simulation of MSF plants. The set of equations was solved in a global manner by arranging the stage energy relations in the form of a tri-diagonal matrix. This method is stable and shows fast convergence. Theoretical models which simulate transient behavior of MSF plants under various conditions have also been reported in the literature [Rimawi et al., 1989; Aly and Marwan, 1995]. Generally, the models were based on coupling the dynamic equations of mass and energy for brine and product tray within the flash stages.

4.2 MSF Desalination System Description

A brief description of a MSF desalination process is provided in this section. The MSF process includes a number of flashing stages connected to a brine heater. The flashing stages are

composed of two sections - the heat rejection section and the heat recovery section, as shown in Figure 4.1. The feed to the plant, W_T , is allowed to pass through the heat rejection section, whose function is to reject the surplus thermal energy from the plant and to cool the product and brine to the lowest possible temperature as they emerge from the last (coolest) rejection stage.

On leaving the first (warmest) rejection stage the feed stream is split into two parts, reject seawater C_w , which passes back to the sea and a make-up stream F which is then combined with the recycle stream R . The combined stream W_F now passes through a series of heat exchangers in the heat recovery section. The temperature of the combined stream rises as it proceeds towards the heat input section of the plant. Passing through the brine heater, the brine temperature is raised from T_{F_1} at the inlet of the brine heater to the top brine temperature (TBT) T_{B_0} approximately equal to the saturation temperature at the system pressure.

The brine then enters the first heat recovery stage through an orifice thus reducing the pressure. As the brine was already at its saturation temperature for a higher pressure, it will become superheated and flashes to give off water vapor. This vapor passes through a wire mesh (demister) to remove any entrained brine droplets and on to a heat exchanger where the vapor is condensed and drips into a distillate tray. The process is then repeated all the way down the plant as both brine and distillate enter the next stage which is at a lower pressure. The concentrated brine is divided into two parts as it leaves the plant, the blowdown BD which is pumped back to the sea and a recycle stream R which returns to mix with the make-up stream, F .

From the mathematical point of view, the once-through design and the recirculation design can be represented by the same model if the flow rates of the recycle R and the reject seawater C_w streams are given zero values. Moreover, there will be no distinction between heat recovery and heat rejection sections in the case of the once-through plant.

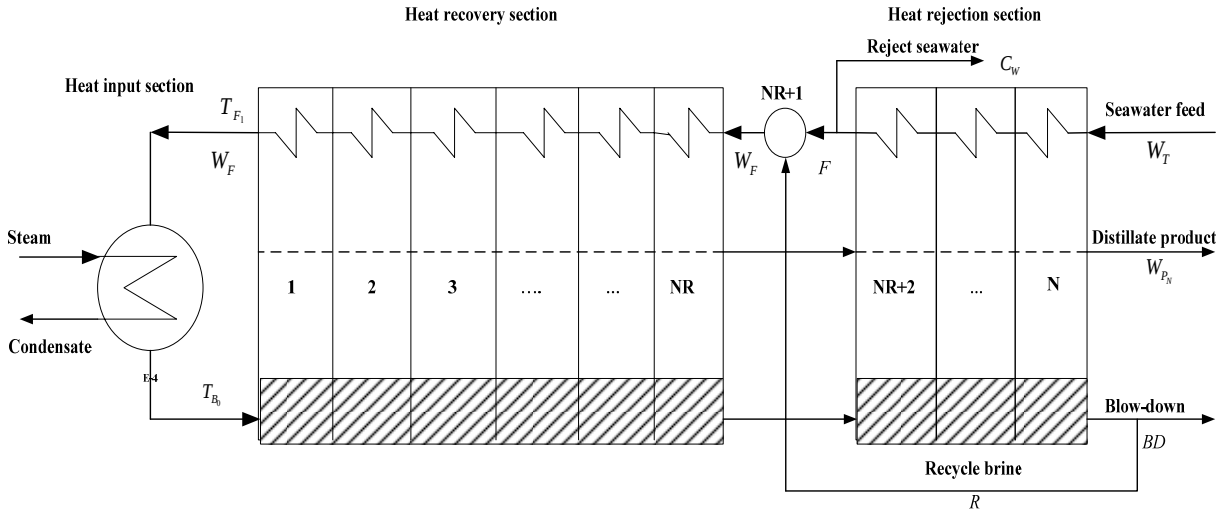


Figure 4.1. Schematic diagram of a MSF process with brine recirculation.

4.3 MSF Mathematical Models

In order to develop the mathematical models of a multi-stage flash desalination process, the following simplifying assumptions are made [Thomas et al., 1998]:

- The distillate product leaving any stage is salt free
- The heats of mixing for brine solutions are negligible
- No mist or salt is entrained with the vapor
- There is no heat loss from the plant
- The condensate produced in the brine heater is not sub-cooled and flows through the brine heater at a constant temperature
- The accumulation of cooling brine in the tube of the heat exchangers is neglected since its density changes are negligible
- The MSF plant is divided into a brine heater and a number of adiabatic stages while the interaction among them is only through the flow streams connecting them.
- The pumps are not explicitly considered in the model. It is assumed that the change in flow rate in the line due to valve action is transmitted to all parts of the flow line instantaneously.
- The model was developed using lumped parameter analysis, the mass considered to be perfectly mixed and spatial variations were not explicitly considered.

4.3.1 Flashing Stage Dynamic Model

A typical MSF flashing stage can be divided into four control volumes (CV): the flashing brine tray, the distillate (product) tray, the vapor space, and the condenser tubes [Thomas et al., 1998]. Figure 4.2 shows the block diagram of a single stage. It should be noted that the masses

are lumped at the exit of the CV and hence the exit temperatures T_{B_i}, T_{P_i} , and T_{F_i} are well representative of the temperatures of their corresponding CVs.

The mass balance equation of the flashing brine tray:

$$\frac{dM_{B_i}}{dt} = W_{B_{i-1}} - W_{B_i} - W_{Bf_i} \quad (4.1)$$

The balance of salt mass:

$$\frac{d(M_{B_i} X_{B_i})}{dt} = W_{B_{i-1}} X_{B_{i-1}} - W_{B_i} X_{B_i} \quad (4.2)$$

The energy balance of the flashing brine tray:

$$\frac{d(M_{B_i} h_{B_i})}{dt} = W_{B_{i-1}} h_{B_{i-1}} - W_{B_i} h_{B_i} - W_{Bf_i} h_{Bf_i} \quad (4.3)$$

The mass balance of the product tray:

$$\frac{dM_{P_i}}{dt} = W_{P_{i-1}} - W_{P_i} - W_{Pf_i} + W_{C_i} \quad (4.4)$$

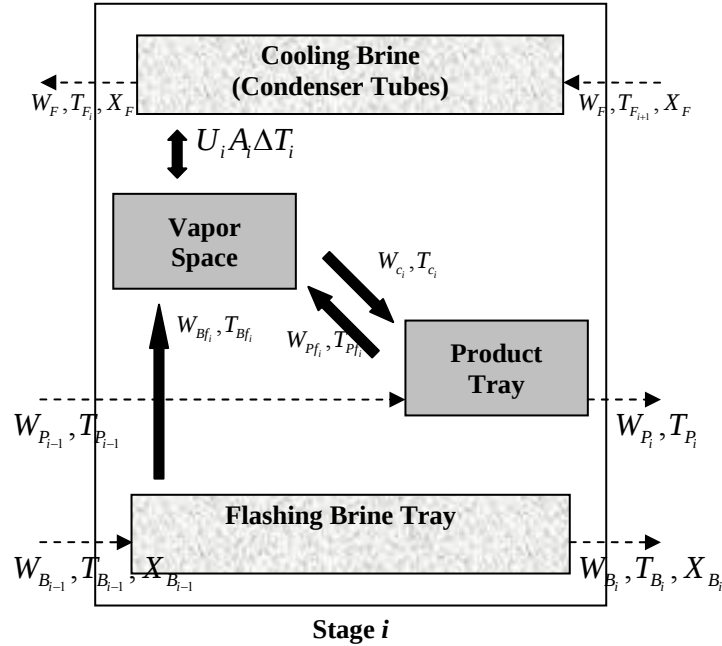


Figure 4.2. Block diagram of a generic MSF stage.

The energy balance of the product tray:

$$\frac{d(M_{P_i} h_{P_i})}{dt} = W_{P_{i-1}} h_{P_{i-1}} - W_{P_i} h_{P_i} - W_{P_{fi}} h_{P_{fi}} + W_{C_i} h_{C_i} \quad (4.5)$$

The mass balance of the vapor space:

$$\frac{dM_{V_i}}{dt} = W_{B_{fi}} + W_{P_{fi}} - W_{C_i} \quad (4.6)$$

The energy balance of the vapor space:

$$\frac{d(M_{V_i} h_{V_i})}{dt} = W_{B_{fi}} h_{B_{fi}} + W_{P_{fi}} h_{P_{fi}} - W_{C_i} h_{C_i} - U_i A_i \Delta T_i \quad (4.7)$$

where U_i is the heat transfer coefficient, A_i is the heat transfer area, and ΔT_i is the log mean temperature difference (LMTD), given by:

$$\Delta T_i = \frac{T_{F_i} - T_{F_{i+1}}}{\ln \left(\frac{T_{P_i} - T_{F_{i+1}}}{T_{P_i} - T_{F_i}} \right)}$$

The energy balance of the condenser tubes:

$$M_{F_i} \frac{dh_{F_i}}{dt} = W_F (h_{F_{i+1}} - h_{F_i}) + U_i A_i \Delta T_i \quad (4.8)$$

The distillate temperature will be less than the brine temperature by the boiling point elevation (BPE), non-equilibrium allowance (NEA) and loss in saturation temperature due to other pressure losses (Δ):

$$T_{B_i} = T_{P_i} + BPE_i + NEA_i + \Delta_i \quad (4.9)$$

The saturation temperature of the vapor and distillate temperature correlation:

$$T_{V_i} = T_{P_i} + \Delta_i \quad (4.10)$$

The relationship for the evaluation of the stage pressure f_1 and empirical relationships for the evaluation of the flashing brine and distillate flow rates f_2 and f_3 (details in Appendix A):

$$P_i = f_1(T_{V_i}) \quad (4.11)$$

$$W_{B_i} = f_2(P_i, P_{i+1}, M_{B_i}, M_{B_{i+1}}) \quad (4.12)$$

$$W_{P_i} = f_3(P_i, P_{i+1}, M_{P_i}, M_{P_{i+1}}) \quad (4.13)$$

It is assumed the vapor from the brine and distillate tray are not superheated, the condensate is not sub-cooled, and the vapor space has fully equilibrated with the stage conditions, that is,

$$T_{C_i} = T_{B_{f_i}} = T_{P_{f_i}} = T_{V_i}$$

4.3.2 Brine Heater Dynamic Model

The brine heater (BH) has a steam flow, W_{STEAM} , coming in, which is considered to be saturated. BH is considered as the 0th stage; hence the subscript 0 is used [Thomas et al., 1998].

Enthalpy balance of the cooling brine:

$$M_{B_0} \frac{dh_{B_0}}{dt} = U_{BH} A_{BH} \Delta T_{BH} - W_F (h_{B_0} - h_{F_1}) \quad (4.14)$$

where U_{BH} is the heat transfer coefficient, A_{BH} is the heat transfer area, and ΔT_{BH} is given by:

$$\Delta T_{BH} = \frac{T_{B_0} - T_{F_1}}{\ln \left(\frac{T_{STEAM} - T_{F_1}}{T_{STEAM} - T_{B_0}} \right)}$$

Enthalpy balance of the condensing steam:

$$W_{STEAM} \lambda_{STEAM} = U_{BH} A_{BH} \Delta T_{BH} \quad (4.15)$$

4.3.3 MSF Steady-State Model

The mathematical model described in the previous section consists of a system of ordinary differential equations combined with algebraic equations. The differential equations are linear with respect to the time derivatives of the unknown variables of the model. In the steady-state case, the time derivatives are set to zero. The mass hold-up is no longer a variable in any equation. The enthalpy in the equations is written in terms of the relevant temperature.

Enthalpy balance of the flashing brine tray:

$$W_{B_{i-1}} h_{B_{i-1}} = W_{B_i} h_{B_i} + (W_{B_{i-1}} - W_{B_i}) h_{Bf_i} \quad (4.16)$$

Total material balance:

$$W_{B_{i-1}} + W_{P_{i-1}} = W_{B_i} + W_{P_i} \quad (4.17)$$

Salt balance of the flashing brine tray:

$$W_{B_{i-1}} X_{B_{i-1}} = W_{B_i} X_{B_i} \quad (4.18)$$

Overall enthalpy balance of a stage:

$$W_F C_{p_{AV_i}} (T_{F_i} - T_{F_{i+1}}) = W_{P_{i-1}} C_{p_{P_{i-1}}} (T_{P_{i-1}} - T^*) + W_{B_{i-1}} C_{p_{B_{i-1}}} (T_{B_{i-1}} - T^*) \\ - W_{P_i} C_{p_{P_i}} (T_{P_i} - T^*) - W_{B_i} C_{p_{B_i}} (T_{B_i} - T^*) \quad (4.19)$$

Heat transfer equation:

$$W_F C_{p_{AV_i}} (T_{F_i} - T_{F_{i+1}}) = U_i A_i \frac{T_{F_i} - T_{F_{i+1}}}{\ln \left(\frac{T_{P_i} - T_{F_{i+1}}}{T_{P_i} - T_{F_i}} \right)} \quad (4.20)$$

Temperature correction equation:

$$T_{B_i} = T_{P_i} + BPE_i + NEA_i + \Delta_i \quad (4.21)$$

Brine heater overall enthalpy balance equation:

$$W_F C_{p_{F_{BH}}} (T_{B_0} - T_{F_1}) = W_{STEAM} \lambda_{STEAM} \quad (4.22)$$

Heat transfer equation of brine heater:

$$W_F C_{p_{F_{BH}}} (T_{B_0} - T_{F_1}) = U_{BH} A_{BH} \frac{T_{B_0} - T_{F_1}}{\ln \left(\frac{T_{STEAM} - T_{F_1}}{T_{STEAM} - T_{B_0}} \right)} \quad (4.23)$$

For steady-state solution of the MSF plant model, all the input variables (operating conditions), the plant dimensions, and the physical properties of the streams of the plant should be known. The mathematical expressions describing the thermo-physical properties of water, steam, and brine solutions are given in Appendix A. It is noteworthy that the functions

representing those properties are highly nonlinear, thus contributing a lot to the overall complexity of the model. A stage-by-stage approach is usually highly iterative and is characterized by slow convergence. Therefore, a global approach presented by Helal et al. [1986] is employed in this study. This approach arranges the stage energy equations of all the stages in a TDM form, and is capable of generating stable results with fast convergence.

4.4 Control Loops in a MSF System

The main control loops of a MSF process are shown in Figure 4.3. Each of the control loops involves a single-input single-output system. The controlled and manipulated variables include the following:

(1) Temperature of the intake seawater entering the last flashing stage (controlled) and the circulation flow rate of the warm cooling seawater (manipulated). This loop operates only during the winter season when the intake seawater temperature drops to values close to 15°C . As a result, the control valve and pumping system of the control loop operates to adjust the temperature of the intake seawater. Set-points for the intake seawater temperature vary between 25°C for winter operation and 32°C for summer conditions. This loop is not found in the once through MSF layout [El-Dessouky et al., 1999].

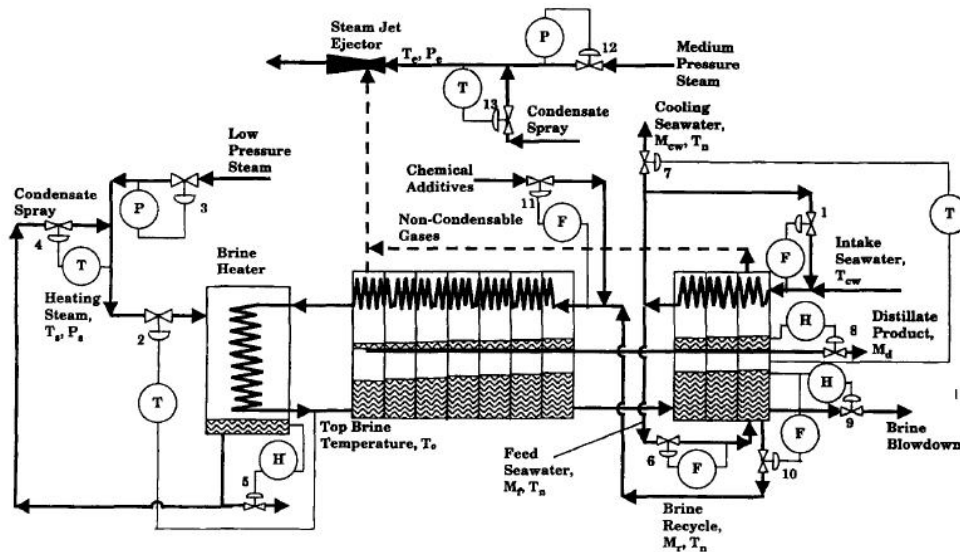


Figure 4.3. Single-input single-output MSF control loops [Alatiqi et al., 1999].

(2) TBT (controlled) and heating steam flow rate (manipulated). An increase or decrease of the heating steam flow rate is necessary to control the TBT. This might be necessary to take into account an increase in the brine circulation rate, fouling and scaling effects, or decrease in the temperature of the brine circulation. The set point for the TBT depends on the type of antiscalant used in the plant where 90°C is suitable for a polyphosphate type and $100\text{-}110^{\circ}\text{C}$ is used for polymer type additives.

(3) Pressure of the heating steam (controlled) and the opening of the throttling valve (manipulated). Commonly, the low pressure steam has a pressure of 4-7 bars, and it is necessary to reduce its pressure to a value of about 2 bar where it becomes superheated with a temperature close to 170°C .

(4) Temperature of the heating steam (controlled) and the flow rate of the condensate spray (manipulated). This loop changes the steam quality from superheated to saturated, where its temperature drops from 170°C to 100 or 110°C . The steam temperature is controlled to be higher than the TBT by $5\text{-}10^{\circ}\text{C}$. This is necessary to prevent the formation of hot spots and scale formation.

(5) Condensate level in the brine heater (controlled) and the opening of the distillate discharge valve (manipulated). It is necessary to maintain a sufficient static head above the condensate pump to prevent condensate flashing within the discharge line or the pumping unit which would result in violent vibrations, loosening of various connections, and severe erosion. This loop, like all other level loops in the plant, is consistent with the material balance in direction of flow principle.

(6) The flow rate of the feed seawater entering the brine pool of the last stage (controlled) and the valve opening on this line (manipulated). The set point for this control is the flow rate ratio of distillate and brine blow down. This control loop maintains a constant conversion ratio, which is necessary to keep the salinity of the brine blow-down from the last stage at a design value of 70,000 ppm. In the Gulf, the intake seawater salinity varies between 42,000-48,000 ppm

and accordingly, the conversion ratio varies between 0.4-0.314. This sets the controller ratio at 1.5-2.18, with the units of kg brine blow-down per 1 kg distillate product.

(7) Temperature of the reject flow rate of the cooling seawater (controlled) and the opening of the discharge valve (manipulated). The set-point of the controller is made in reference to the temperature of the last flashing stage. For a constant plant capacity, increase in the stage temperature causes reduction in the flashing range and the system performance ratio with simultaneous increase in the brine recycle flow rate. The last stage temperature is set at 40°C for summer operation and 32°C for winter operation.

(8) Distillate level in the last stage (controlled) and the distillate product flow rate (manipulated). This control loop has a similar function to the level controller of the heating steam condensate where decrease of the distillate product static head would cause operational problems to the associated pumping unit.

(9) Brine level in the last flashing stage (controlled) and the brine blow-down flow rate (manipulated). This is one of the most important control loops in a MSF plant since the head in the last stage adjusts the head in previous stages. Simultaneously, the flashing efficiency is affected in various stages as well as the amount of distillate product. Increase of the head reduces the flashing rates and the amount of distillate product; consequently, the amount of heat released upon condensation of the distillate vapor becomes lower. Eventually, the temperature of the brine recycle stream entering the brine heater becomes lower, which results in the increase in the amount of heating steam and reduction in the system thermal performance ratio. On the other hand, reduction of the brine head may result in blow-through of the flashed-off vapors across the stages. As a result, the effective number of stages becomes smaller and the plant performance ratio decreases.

(10) Brine level in the last flashing stage (controlled) and the brine recycle flow rate (manipulated). This control loop has a similar function and effects on the system performance as the previous control loop.

(11) Dosing flow rate of the chemical additives (controlled) and the opening of the discharge valve of the chemical additive (manipulated). Variations in the brine circulation flow rate necessitate adjustment of the dosing rate of the chemical additives. Such variations are caused by adjustments in the flow rate of the brine circulation stream which might be necessary to adjust the brine level in the last flashing stage.

(12) Pressure of the ejector motive steam (controlled) and the opening of the throttling valve (manipulated). Commonly, the medium pressure steam has a pressure of 16 bars and it is necessary to reduce its value to 7 bars.

(13) Temperature of the ejector motive steam (controlled) and the flow rate of the condensate spray (manipulated). This loop reduces the ejector motive steam temperature from 201.4°C to 165°C. Spray of steam condensate results in this temperature reduction.

Selection of the MSF controllers is made subject to the following classification: the controller proportional action works on the set point changes and the controller derivative action does not work on set point changes. This is necessary to obtain a fast response and to limit system disturbances.

4.5 Measurements and Instrumentation in a MSF System

Instrumentation and measurements form an essential part in various control loops in desalination plants where measured signals are transmitted to the control system and compared against the desired set points. Subsequently, corrective action in various manipulated parameters takes place to adjust the controlled variable to the desired value. Accordingly, maintaining measuring instrumentation in good working condition requires frequent cleaning, servicing, and calibration. Alatiqi et al. [1999] outlined in detail the main measuring instruments in MSF plants. A summary of these instruments is given in Table 4.1. All of the temperature, pressure, flow rate, level, and pH measurements are made on-line, while the measured data are taken at equal intervals and logged on a data acquisition system. Other data, such as, detailed water analysis, are made off-line in central analytical laboratories following standard methods. As is shown, some measuring instruments are simple and inexpensive to use; for example, differential

Table 4.1. Measuring instruments used in the desalination industry

Measured parameter	Typical problems
Temperature	1. Small temperature differences. Measuring error can be in the same range as the temperature difference between two stages. 2. Two phase temperature measurements yield a high error due to vapor condensation and presence of non-condensable gases. 3. Temperature non-uniformity, measurement may alter due to presence of gradients or heat losses. 4. High corrosion rates of sensors, wires, and cables in presence of high temperature seawater and brine.
Pressure	1. Vapor condensation in the tubing leading to the sensing sides of the transducers. 2. Non-condensable gases may generate large errors. 3. Pressure variations along the circumference of pipes. 4. Tap fouling and blockage gives partial readings of pressure and large errors.
Flow	1. Errors caused by bends, expansion, and contraction. 2. Flashing effects gives erroneous errors for flow meters with moving parts. 3. Fouling and scaling may hinders motion of moving parts and block small openings.
Level	1. Filling of the vapor/gas leg with condensate may result in large errors. 2. Accumulation of dirt in the liquid leg may generate erroneous errors. 3. Oscillation effects due to violent flashing effects or turbulence.
pH	1. Fouling problems around sensors. 2. Poor reading due the delicate nature of the sensor or bad mixing.
Gas concentration	1. Fouling problems. 2. Calibration problems.
Water composition	1. Protect sensor from fouling. 2. Poor mixing and dead zones. 3. Delicate nature of sensors.

manometers. However, a major part of the instrumentation is delicate and requires frequent calibration and replacement of defective parts, such as, pH and conductivity probes. Also, all of the measuring instrumentation may face problems related to fouling, poor mixing, vapor condensation, and the corrosive nature of the seawater. In various locations around the plant, multiple sensors might be used to obtain an average value or ensure proper measurements of sensitive parameters, i.e., TBT, brine level in the last stage, or pressure and temperature of heating steam.

All measuring instruments are connected with indicators and alarms commonly located in the central operation and control room. Some instrumentation is tied with a control loop; for example, TBT, brine level in the last stage, and pressure of the heating steam. A small number of the instruments are coupled with an alarm system; this may vary from one plant to another. However, in all plants the TBT is tied with an alarm system. Other alarm units are used for the distillate flow rate, distillate level in the last stage, brine level in the last stage, brine level in the first stage, and temperature and flow rates of intake and feed seawater.

4.6 MSF Desalination System Simulation Results

Steady-state and dynamic behavior of a MSF plant can be predicted from the mathematical model presented in the previous section. However, the model has been derived under certain simplifying assumptions. Thus, it is necessary to validate the given model, obtained from the design data, against the actual plant operation data. Steady-state behavior of the MSF model is presented first.

4.6.1 Steady-State Simulation Results

Assuming that stage and brine heater geometries are assigned, together with the values of the fouling factors in all the heat exchangers of the plant, the feed temperature T_{SEA} and concentration X_{SEA} , and comparing the number of equations and the number of unknowns of the model allows to determine that the developed model has four degrees of freedom. Below are

three different combinations of variables that can be chosen to simulate a MSF plant. Each set comprises four specifications as has been explained before.

(1) The first case is termed as a “performance” calculation. In this case, the following operating variables are specified: R , C_w , F , T_{STEAM} .

(2) In the second case the operating parameters F , C_w , T_{B_0} and the plant capacity D_N are specified. This study may be used to investigate the possibility of maintaining a specified plant capacity in the condition of changing feed temperature.

(3) The third case deals with the specified parameters F , C_w/R , W_{STEAM} and T_{B_0} . This alternative simulates a situation where a specified amount of steam is supplied to the desalination plant by a coupled power plant.

Here we consider performance calculation, where we specify the make-up flow, F , the rejected seawater flow rate C_w , the recycle stream flow rate R , and the steam temperature T_{STEAM} . On the basis of the information the model allows to calculate the profiles of temperatures and flow rates in all the stages of the unit.

The configuration investigated in this work refers to the design data of a real desalination plant located in the Middle East which includes 13 stages in the heat recovery section and 3 heat rejection stages. The plant design data are given in Tables 4.2 - 4.4.

In the performance calculation, the feed seawater temperature and salt concentration are $95^\circ F$ and 5.6% wt, respectively, while the steam temperature, T_{STEAM} , is $206^\circ F$. The other operating parameters required for the calculation are defined as follows:

$$F = 0.125 \times 10^8 \text{ lb/hr}, \quad C_w = 0.124 \times 10^8 \text{ lb/hr}, \quad R = 0.140 \times 10^8 \text{ lb/hr}.$$

The results of the performance calculation are reported in Table 4.5. In particular, the values of the flow rates, temperatures and salt concentrations are given. For the sake of comparison, the actual plant values for the principal parameters at operating conditions are given in Table 4.6. It can be observed that the stage temperatures, inter-stage flow rates, and salt concentrations are predicted very well by the steady-state model.

Table 4.2. The main characteristics of the plant

Type of plant	Brine recycle-cross tube
No. of recovery stages	13 (Stages 1~13)
No. of rejection stages	3 (Stages 14~16)
No. of stages	16
Nominal capacity	5 MGD

Table 4.3. The operating conditions

Steam temperature to brine heater	206 °F
Feed Seawater temperature	95 °F
Feed Seawater concentration	56,000 ppm
Make-up flow rate	0.125×10^8 lb / hr
Recycle flow rate	0.140×10^8 lb / hr
Reject cooling seawater	0.124×10^8 lb / hr
Height of brine level in each stage	18 in.

Table 4.4. The condenser tube bundle data

Parameter	Recovery section	Rejection section	Brine heater
No. of tubes per stage	4300	4300	4300
Tube length (ft)	40	35	40
Inside diameter (in.)	0.866	0.9394	0.866
Thickness (in.)	0.048	0.0303	0.048
Stage width (ft)	40	35	40
Thermal conductivity (BTU / hr · ft · °F)	28.9	9.4	16.8
Fouling factor (BTU / hr · °F · ft ²) ⁻¹	0.681×10^{-3}	0.1136×10^{-3}	0.909×10^{-3}
Heat transfer area (ft ²)	43,000	38,000	38,000

Table 4.5. Steady-state model results corresponding to the performance study case

Stage #	W_{B_i} ($\times 10^8 \text{ lb/hr}$)	W_{P_i} ($\times 10^6 \text{ lb/hr}$)	X_{B_i} (%)	T_{F_i} ($^{\circ}\text{F}$)	T_{P_i} ($^{\circ}\text{F}$)	T_{B_i} ($^{\circ}\text{F}$)	U_i ($\text{BTU/hr} \cdot \text{ft}^2 \cdot ^{\circ}\text{F}$)
0 brine heater	0.2650	0	6.2412	0	0	200.50	696.7
1	0.2636	0.1420	6.2749	190.23	192.86	194.97	676.7
2	0.2621	0.2862	6.3094	184.46	187.16	189.28	674.9
3	0.2607	0.4324	6.3448	178.55	181.31	183.45	673.2
4	0.2592	0.5802	6.3810	172.51	175.34	177.50	671.6
5	0.2577	0.7293	6.4179	166.36	169.24	171.43	670.2
6	0.2562	0.8792	6.4554	160.10	163.03	165.25	668.9
7	0.2547	1.0296	6.4935	153.76	156.72	158.99	667.7
8	0.2532	1.1800	6.5321	147.34	150.33	152.66	666.6
9	0.2517	1.3301	6.5711	140.85	143.87	146.28	665.5
10	0.2502	1.4792	6.6102	134.33	137.36	139.86	664.4
11	0.2487	1.6270	6.6495	127.79	130.83	133.44	663.1
12	0.2473	1.7728	6.6887	121.25	124.28	127.02	661.5
13	0.2458	1.9161	6.7277	114.73	117.75	120.66	659.6
14	0.2448	2.0183	6.7558	110.01	112.90	116.08	614.4
15	0.2438	2.1216	6.7844	105.09	108.05	111.40	611.4
16	0.2427	2.2256	6.8135	100.08	103.10	106.67	607.9

Table 4.6. Actual plant operation data [Helal et al., 1986]

Stage #	W_{B_i} ($\times 10^6 \text{ lb/hr}$)	W_{P_i} ($\times 10^6 \text{ lb/hr}$)	X_{B_i} (%)	T_{F_i} ($^{\circ}\text{F}$)	T_{P_i} ($^{\circ}\text{F}$)	T_{B_i} ($^{\circ}\text{F}$)	U_i ($\text{BTU/hr} \cdot \text{ft}^2 \cdot ^{\circ}\text{F}$)
0 brine heater	0.2650	0	6.2412	0	0	199.35	696.7
1	0.2633	0.1711	6.39	187.46	190.53	192.66	675.2
2	0.2614	0.3440	6.44	180.74	183.66	185.80	673.3
3	0.2600	0.5018	6.47	174.38	177.29	179.47	671.6
4	0.2584	0.6581	6.51	168.01	170.92	173.11	670.1
5	0.2569	0.8116	6.55	161.70	164.57	166.81	668.8
6	0.2554	0.9615	6.59	155.45	158.30	160.58	667.7
7	0.2539	1.1079	6.63	149.27	152.09	154.43	666.6
8	0.2525	1.2509	6.67	143.16	145.95	148.36	665.7
9	0.2511	1.3906	6.70	137.12	139.88	142.36	664.7
10	0.2497	1.5270	6.74	131.13	133.87	136.45	663.6
11	0.2484	1.6602	6.78	125.22	127.93	130.62	662.4
12	0.2471	1.7902	6.81	119.36	122.06	124.87	660.9
13	0.2458	1.9169	6.85	113.57	116.24	119.21	659.1
14	0.2450	1.9980	6.87	109.73	112.36	115.57	641.8
15	0.2440	2.1007	6.90	104.90	107.61	110.92	639.2
16	0.2429	2.2055	6.93	99.96	102.63	106.14	631.9

4.6.2 Dynamic Simulation Results

SIMULINK is a software package for modeling, simulating, and analyzing dynamic systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multi-rate, i.e., have different parts that are sampled or updated at different rates. For modeling, SIMULINK provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. It includes a comprehensive block library of sinks, sources, linear and nonlinear components, and connectors. Models are hierarchical, so models can be built using both top-down and bottom-up approaches. After the model is defined the simulations can be performed using a choice of integration methods, either from the SIMULINK menus or by entering commands in the MATLAB command window. Model analysis tools include linearization and trimming tools, which can be accessed from the MATLAB command line, plus the many tools in MATLAB and its application toolboxes.

A SIMULINK model of a MSF plant with three flash stages and the brine heater is developed in this dissertation, as shown in Figure 4.4. Only blocks from the standard SIMULINK library are used, so that the minimal configuration of SIMULINK is necessary to model a MSF process. Proportional-integral controllers are integrated into the MSF model to regulate the top brine temperature and the brine level in the last flashing stage by controlling the heating steam flow rate through the brine heater and the brine blow-down flow rate, respectively. The set points for the controllers are listed in Table 4.7. Regulation of the brine level in the last flashing stage is one of the most important control loops in a MSF plant, since the head in the last stage adjusts the head in previous stages. Simultaneously, the flashing efficiency is affected in various stages as well as the amount of distillate product. An increase in the head reduces the flashing rates and the amount of distillate product; consequently, the amount of heat released upon condensation of the distillate vapor becomes lower. Eventually, the temperature of the brine recycle stream entering the brine heater becomes lower, which results in the increase in the amount of heating steam and reduction in the system thermal performance ratio. On the other hand, reduction of the brine head may result in blow-through of the flashed-off vapors across the

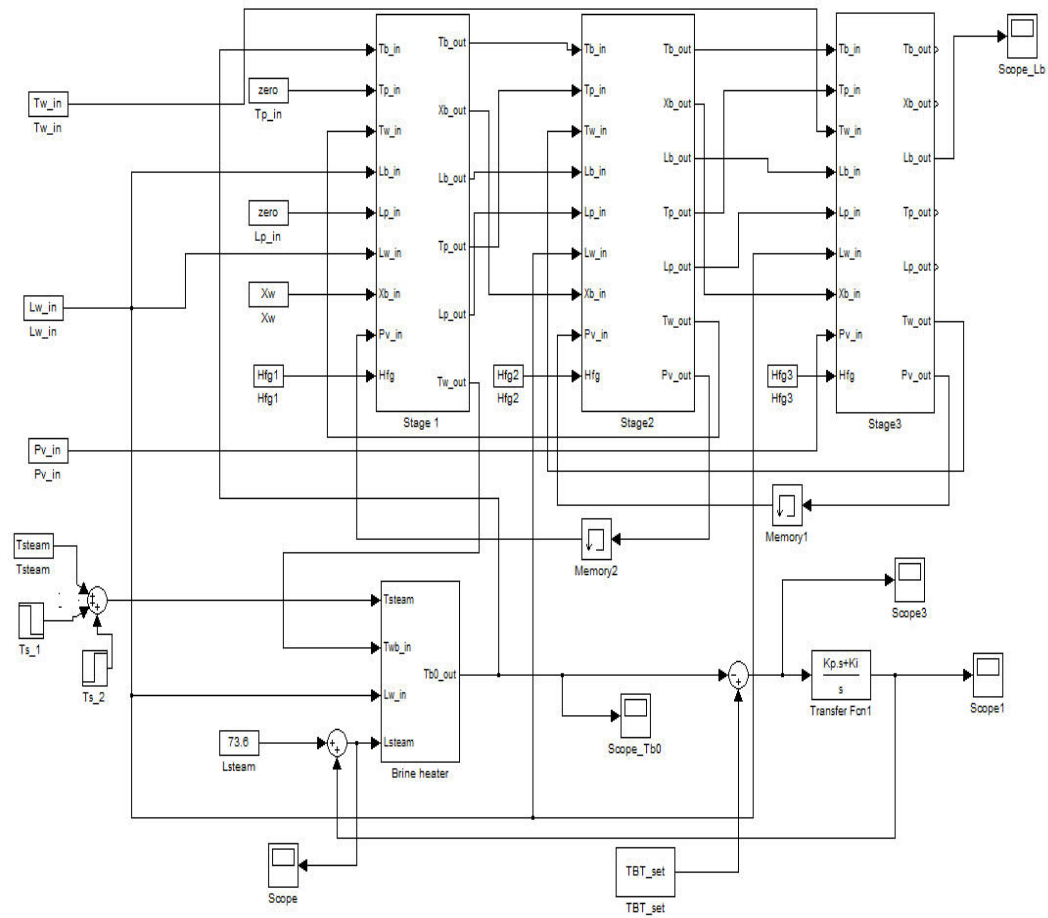


Figure 4.4. Screenshot of MSF SIMULINK models.

Table 4.7. Control loops in dynamic simulation

Control loop	Nature	Set point	Controlled variable	Manipulated variable
1	PI	200.5 °F	Top brine temperature	Steam valve position
2	PI	0.57 m	Last stage brine level	Blow-down valve position

stages. As a result, the effective number of stages becomes smaller and the plant performance decreases.

In dynamic simulation, usually, any one or more of the operator controlled variables are perturbed and its effect on the whole plant is studied. Such perturbations, in real conditions, may also be caused by minor failure of some components, i.e., failures which will not lead to immediate shut down of the plant and which gives a possibility of corrective measures. The main operator controlled variables are steam flow rate through the brine heater, brine recirculation flow rate, sea-water reject flow rate, make-up flow rate, sea-water reject flow temperature at entrance to reject section, last stage brine level and last stage distillate level. The other two main input variables, which are location-specific, are sea-water salinity and steam supply pressure. Table 4.8 summarizes the step changes that are introduced in the key variables, such as the heating steam temperature, recirculation brine flow rate, and recirculation brine temperature. It is important to note that the initial conditions of the dynamic model are set to the steady-state values that have been calculated using the steady-state model presented in the previous section.

Figures 4.5 to 4.8 present the dynamic responses of the top brine temperature, brine heater inlet steam flow rate, and brine levels in all three stages under different dynamic conditions mentioned in Table 4.8. In all the cases, the steady-state simulation is carried out for 100 seconds before introducing step perturbations.

Table 4.8. Dynamic simulation tests

Step change:
2 °F increase in the set point of top brine temperature
10 % increase in the incoming cooling brine flow rate
2 °F increase in the incoming cooling brine temperature
5 °F increase in the heating steam temperature

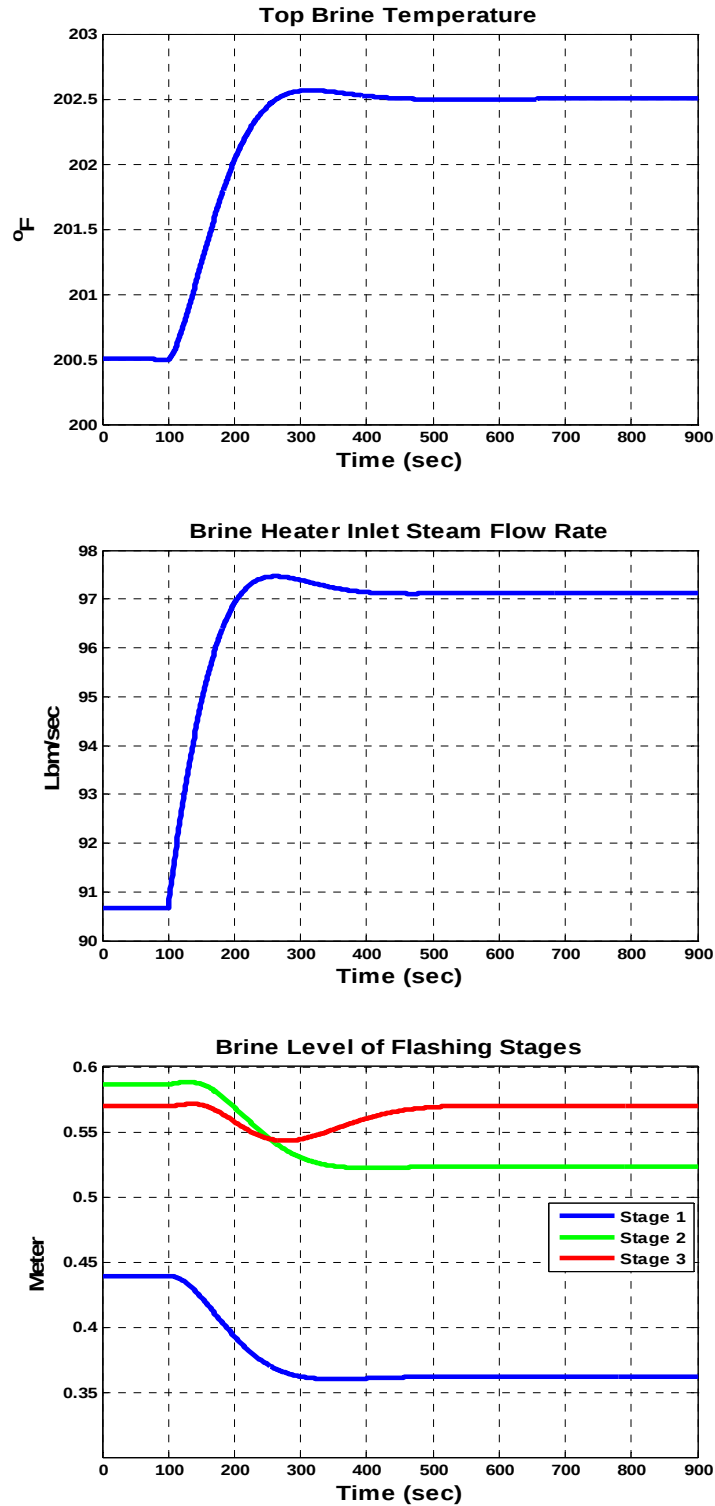


Figure 4.5. Dynamic responses of process variables to step increase of TBT set point.

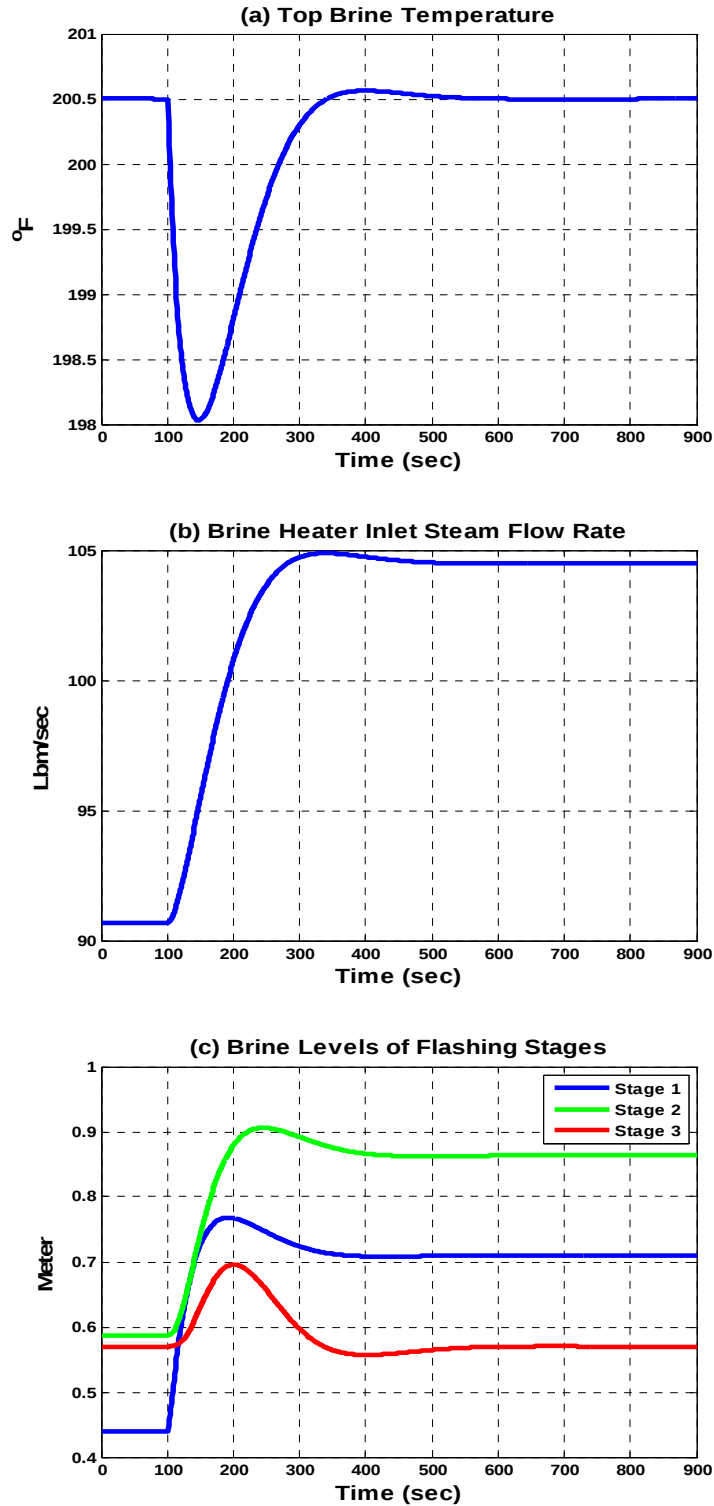


Figure 4.6. Dynamic responses of process variables to step increase of incoming cooling brine flow rate.

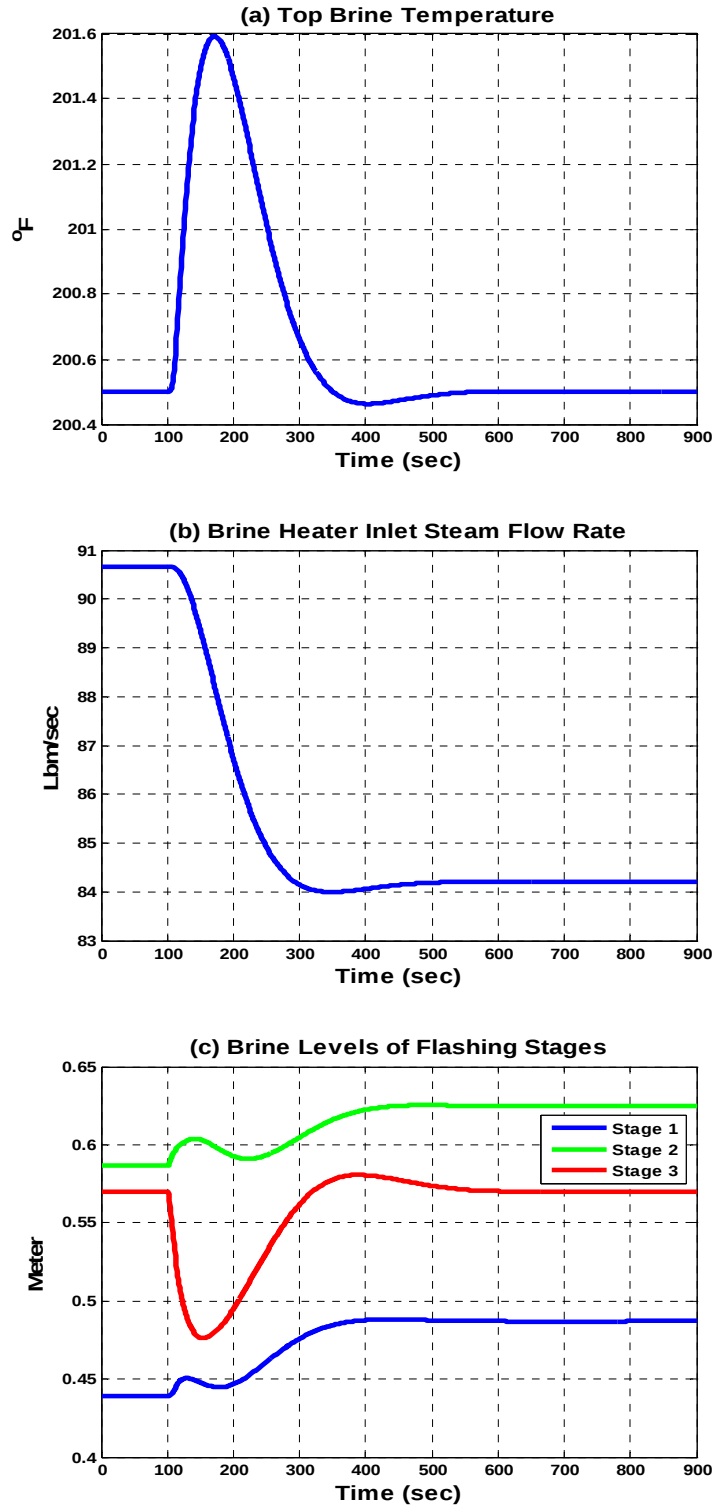


Figure 4.7. Dynamic responses of process variables to step increase of incoming cooling brine temperature.

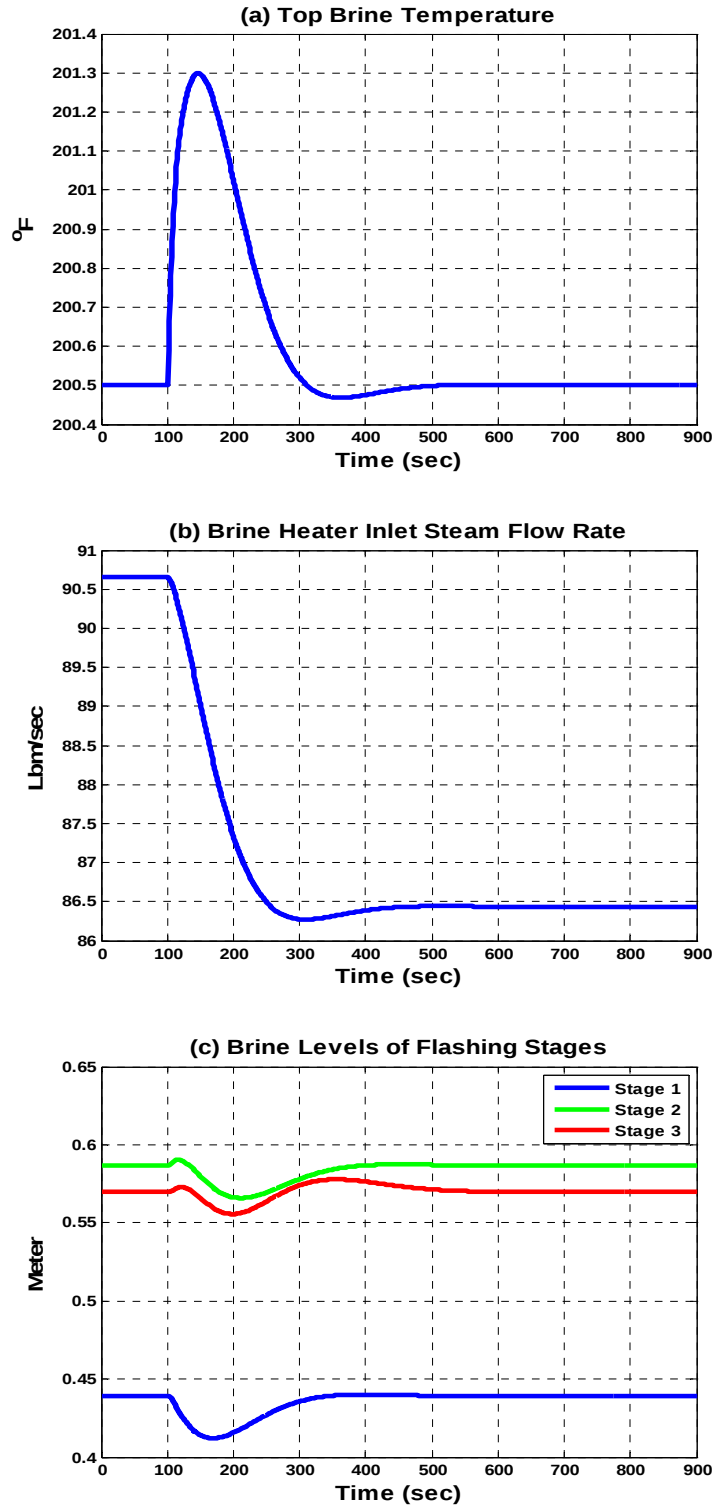


Figure 4.8. Dynamic responses of process variables to step increase of heating steam temperature.

Figures 4.9-4.10 show the open loop responses of the top brine temperature and the brine levels in all three stages when the TBT control loop is open with a step decrease in the heating steam temperature.

It can be seen from the above figures that the steady state is maintained for the first 100 seconds. When the steam temperature is reduced, the heat transferred to the brine gets reduced and hence TBT also decreases as shown in Figure 4.9. This decrease of TBT reduces the amount of vapor flashed from the brine in all stages; hence, the total distillate production also decreases. The decrease of TBT reduces the temperature of the flashing brine and consequently the pressure in the corresponding stages. These changes are transmitted from one stage to the next sequentially. The reduction of pressure in an upstream stage reduces the flow rate through the corresponding orifice. However, as the reduction of pressure has not been transmitted to the stage downstream of the orifice, the outflow of brine from the downstream stage is maintained at the initial value. This will lead to more outflow of brine and hence the decrease in the liquid level in that stage. It can be clearly seen from Figure 4.10 that the decrease in brine level is progressively transmitted from first stage to the third stage. The reduction in brine level is temporary, as the decrease in flashing rates leads to more brine being left behind in the brine tray. Thus a steady increase in the brine levels can be observed. This effect becomes more apparent in Figure 4.11 when the last stage (Stage #3) brine level is not controlled.

4.7 Summary

This chapter presents an overview of modeling and simulation aspects of multi-stage flash desalination plants. Both steady-state and dynamic simulations are discussed. Steady-state models are useful for analyzing the operating conditions of the plant. A mathematical model was obtained by considering the energy and mass conservation equations for the various streams in each section of a MSF plant. The steady-state equations were obtained by setting all the time derivative terms to zero. The steady-state model was simulated for a real plant operating conditions and a close agreement has been observed between the predicted results and the actual plant operating parameters.

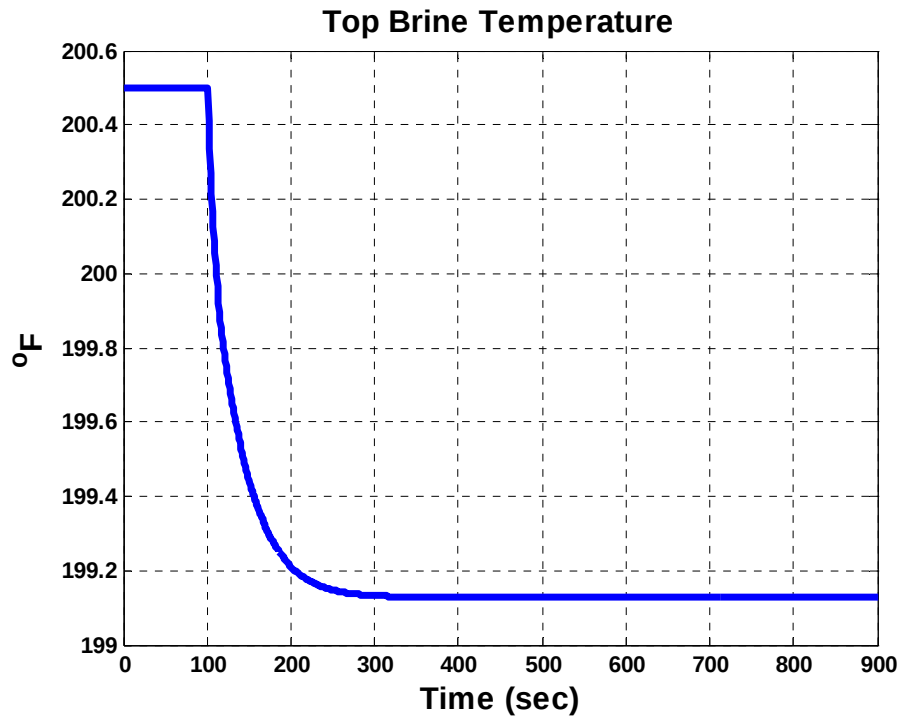


Figure 4.9. Open loop response of TBT to step decrease of heating steam temperature.

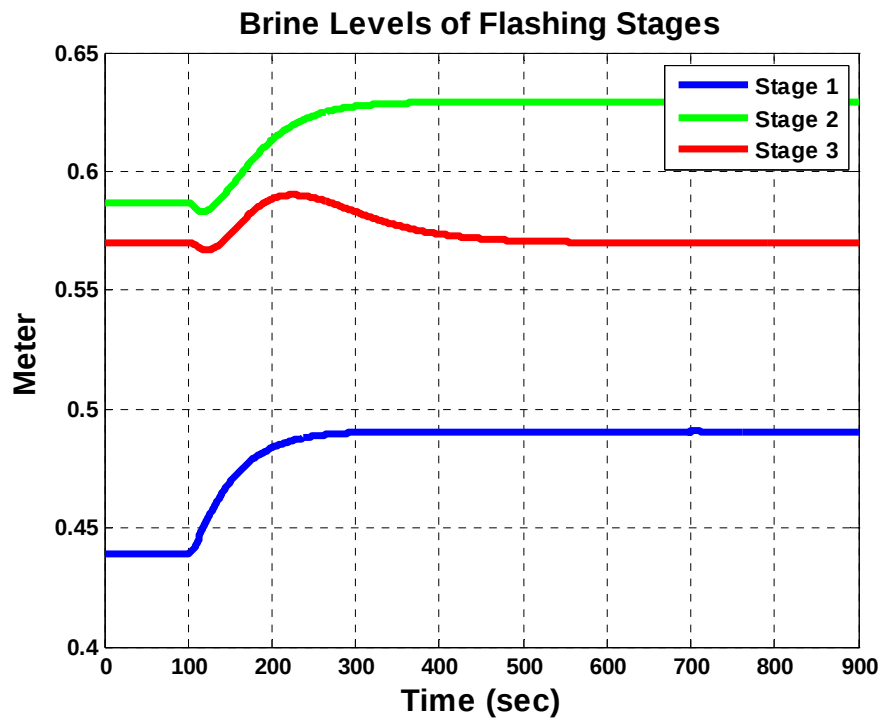


Figure 4.10. Open loop response of brine levels to step decrease of heating steam temperature.

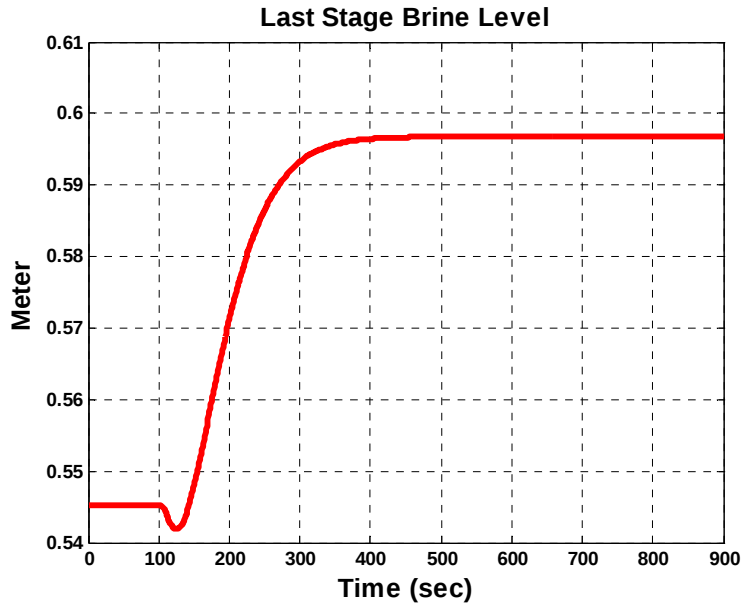


Figure 4.11. Open loop response of last-stage brine level to step decrease of heating steam temperature.

Dynamic models are suitable for solving problems involving transient behavior, such as studying control strategies, stability problems, process interactions, start-up and shut-down conditions. Two types of dynamic models are possible. The first is an analytical one. The analytical model describes the process through physical relations. Essentially, the lumped parameter dynamic model consists of ordinary differential equations and supporting algebraic equations. The initial conditions are either known by experience or developed by steady-state simulation. The other type of dynamic model is based on a black-box approach. A model with unknown parameters is selected according to previous experience or through experiments. Since the formulation of the analytical model is difficult and complicated, it is common to develop black-box models for complex systems. In this dissertation, the former method was employed for MSF dynamic model development. The developed dynamic model was used to simulate the effect of step changes in several key state variables. The simulation results obtained were physically plausible and similar to those found in the literature.

Chapter 5

Modeling and Simulation of a Nuclear Desalination Plant

5.1 Introduction

Nuclear desalination is a viable option as a sustainable source of water and electricity. Nuclear power plants are suitable for large scale desalination application such as MSF, which has been introduced and modeled in Chapter 4. The thermal energy produced in a nuclear plant can provide both electricity and desalted water without the production of greenhouse gases. A particularly attractive option for nuclear desalination is to couple a desalination plant with an advanced, modular, passively safe reactor design, such as the International Reactor Innovative and Secure (IRIS) plant. The design features of the IRIS reactor ensure a safe and reliable source of energy and water even for countries with limited nuclear power experience and infrastructure. In this chapter, the IRIS-MSF nuclear desalination option is explored and analyzed through studying the coupled dynamic SIMULINK models of two plants.

5.2 IRIS System Description

IRIS is one of the next generation nuclear reactor designs developed by an international team of industry, national laboratory, and university partners led by Westinghouse Electric Company [Carelli et al., 2006]. The IRIS design is based on proven light water reactor (LWR) technology, but includes several innovative engineering features that enhance its safety and economics relative to other advanced systems. IRIS is a member of the integral primary system reactor class of designs which houses all functions of the primary coolant system inside the reactor pressure vessel. Figure 5.1 shows the layout of the IRIS primary system.

A unique feature of IRIS is the safety-by-design philosophy which means that design choices are made to eliminate the potential for accidents to occur rather than adding systems to respond to the consequences of accidents. By using an integral system, several potentially severe accident scenarios are avoided, such as medium-to-large pipe break loss-of-coolant accidents

(LOCA). For those accident scenarios that cannot be precluded, the design is such that the consequences of the accident are greatly reduced by the design features. An example is the spherical, high-design-pressure containment vessel that encloses the reactor vessel and safety systems, which is illustrated in Figure 5.2. In the event of a small pipe break in the secondary system, the pressure inside and outside the reactor vessel equalizes very rapidly and prevents the core from being uncovered by coolant.

The novel design features of IRIS are summarized below:

- Scalable in power between 100 MWe and 350 MWe. Baseline design is 335 MWe or 1000 MWt in a modular configuration allowing deployment as single units, multiple single units, or multiple twin units.
- All main primary circuit components (core, control rods and drive mechanisms, steam generators, primary coolant pumps, and pressurizer) are integrated into a single reactor vessel.
- The core is comprised of 89 traditional 17×17-pin PWR fuel bundles containing 4.95% enriched UO₂ fuel and is designed for a 3.5-year cycle with an average burn-up of 50,000 MWd/Te.
- Reactivity control is accomplished through solid burnable absorbers and control rods, and a limited use of soluble boron in the primary coolant.
- The primary coolant system uses eight helical-coil once-through steam generators and eight spool-type coolant pumps. The steam-regulated pressurizer is located in the upper portion of the 6.2-m-diameter, 22.2-m-high reactor vessel. Normal operating pressure of the primary coolant is 15.5 MPa.
- The reactor vessel is placed within a 25-m diameter compact steel containment vessel capable of withstanding 1.4 MPa design pressure.
- All large coolant pipes are eliminated. Only small feed water and steam outlet pipes penetrate the vessel wall (5-cm diameter pipes versus 90-cm-diameter pipes for external loop

PWRs).

- The total inventory of primary coolant is much larger than for an external loop PWR, which increases the heat capacity and thermal inertia of the system and hence yields a much slower response to core heat-up transients.
- The extended riser area provides the possibility for internal placement of the control rod drive mechanisms (CRDM), thus avoiding another potentially serious accident scenario: control rod ejection accident.

Thus, IRIS is designed to fulfill the advantages of the integral primary system reactor. It improves safety, reduces the site civil works, and improves the plant availability for developed and developing countries with large or small electrical grids that can greatly benefit from such design. Thanks to these advantageous design features as well as the specific reasons below, IRIS is well suited for desalination applications.

- The modular sizing of IRIS will allow countries with small-to-medium power requirements to install capacity to their electrical grid in smaller increments and increase additional capacity as their power and water demands warrant and their infrastructure will support.
- The enhanced safety margins will provide additional flexibility in the siting of the reactor to better match the electrical and water use demographics. This will also encourage countries with modest nuclear infrastructure to build and operate nuclear power plants.
- The international nature of the IRIS project team will help to ensure that the design will be licensable for deployment in the global market.

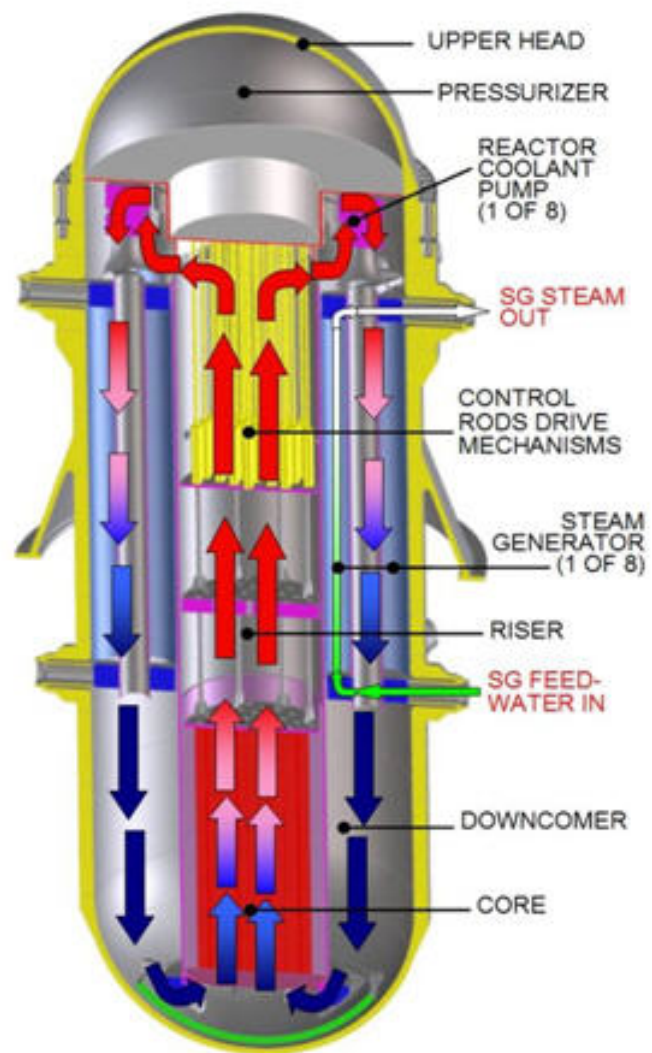


Figure 5.1. IRIS primary system layout [Carelli et al., 2003].

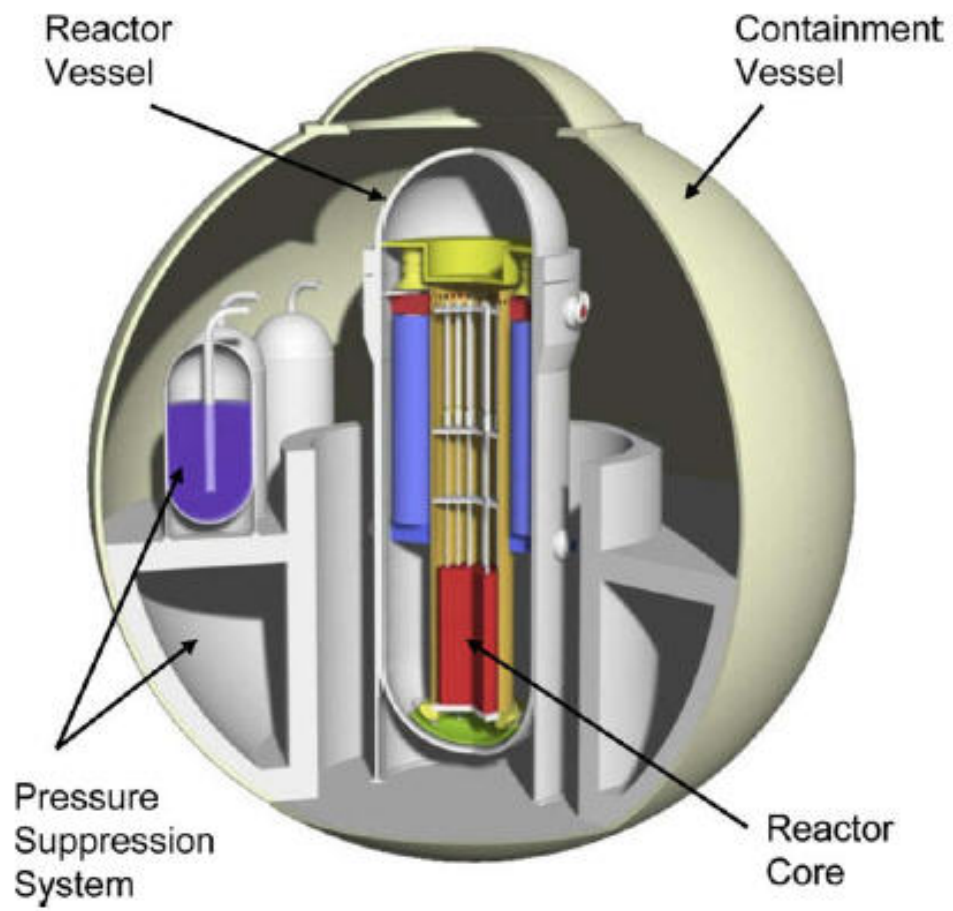


Figure 5.2. IRIS reactor pressure vessel and compact containment vessel [Carelli et al., 2003].

5.3 IRIS Dynamic Models

An IRIS dynamic model developed previously at the University of Tennessee is used in this dissertation, which includes reactor core, helical coil steam generators (HCSG), and balance-of-plant (BOP) systems [Xu, 2009]. The IRIS dynamic models were developed in SIMULINK, and were based on a conventional PWR dynamic model [Naghedolfeizi, 1990] and a HCSG simulation model [Zhao, 2005a]. Major components of the IRIS plant model are described below, and the mathematical equations of the IRIS system are listed in Appendix B.

5.3.1 Reactor Core

The reactor core uses the point reactor kinetics model. The Mann's model is employed to describe the fuel-to-coolant heat transfer in the reactor core. The core model includes the six-group delayed neutron precursor concentration equations, one fuel node temperature, and two coolant node temperature differential equations, along with the fuel and moderator temperature reactivity feedback coefficients. In the reactor core model, the T_{ave} controller is implemented with the fixed set point of $590^{\circ}F$. T_{ave} is defined as the average moderator temperature between the hot leg temperature (T_{hot}) and cold leg temperature (T_{cold}).

5.3.2 Helical Coil Steam Generator

Helical coil steam generator is one of the critical components as well as being the major difference between traditional PWRs and the IRIS reactor. In the HCSG system, the primary fluid enters at the top of the equipment and flows downward to the bottom on the shell side. The primary side heat transfer is sub-cooled forced convection along the entire steam generator height, while the secondary fluid flows upward inside the coiled tubes from bottom to top. The feed water flows into the sub-cooled region of the steam generator, and in this region the heat transfer is mainly due to single-phase turbulent and molecular momentum transfer and the pressure loss is mainly due to wall friction. The saturated region begins when the bulk temperature becomes saturated. The heat transfer in the saturated boiling region is dominated by nucleate boiling, which is much more efficient than single-phase liquid or steam heat transfer. In

the saturated boiling region, the generated bubbles do not disappear in the liquid core and the pressure loss is not only due to the wall friction but also due to the interfacial drag between the bubbles and the liquid. The saturated boiling region ends when the critical heat flux is reached and the liquid film disappears. Due to the relatively large mass flow rate, the critical heat flux occurs at relatively high steam quality. When the steam quality becomes one, the liquid evaporation ceases and the steam becomes superheated.

The size of the steam generator can be reduced through the helical coil design. The heat transfer of the coiled configuration is much more efficient than straight tubes because of the larger heat transfer area per unit volume and the secondary flow induced by the coil geometry. The produced superheated steam also avoids the need of a steam-water separator inside the steam generator. The possibility of tube rupture can be reduced because the secondary fluid flows inside the SG tubes and thus the tubes experience compression force from the outside. The HCSG system control objective is to supply adequate amount of steam to meet the turbine demand, while maintaining the steam pressure at 841 PSI. Therefore, the HCSG model has a proportional-integral steam pressure controller with the fixed set point of 841 PSI, which regulates the steam flow rate through the turbine throttle valve.

The once-through HCSG dynamic model and the steam pressure controller equations can be found in [Zhao, 2005a].

5.3.3 Steam Turbine

A steam turbine, found in a fossil power plant, is one in which the thermal energy of the supplied steam is converted to mechanical energy on the turbine shaft. High pressure and temperature steam, produced in the steam generator, is supplied to the steam turbine as a working fluid. The steam passing through the steam turbine is expanded and thus generates mechanical energy on the turbine shaft. The turbine drives an electric generator to produce electricity. It is noted that a portion of the steam is bypassed to a feedwater heater to improve the thermal efficiency of the turbine cycle.

A dynamic model of a turbine cycle has been developed previously by Shankar [1977]. Modifications have been made to reduce the model complexity. In this dissertation, a simple one-stage turbine model is used for coupling with the condenser and feedwater heater models, which are described next.

5.3.4 Main Condenser

A condenser is a large surface-type heat exchanger, which condenses steam from the exhaust of turbine by transferring steam latent heat to circulating water inside the condenser. The main condenser is equipped with many auxiliary systems such as vacuum and hotwell pumps. The vacuum system maintains a constant pressure in the condenser for transient and steady-state conditions. The hotwell pumps and their control systems discharge the water to a low-pressure heater and control the water level in the hotwells. The condenser is desired to work under vacuum condition to obtain a maximum mechanical power from the turbine system.

The main condenser is simulated as an equilibrium two-phase tank, as shown in Figure 5.3. The space inside the tank is divided into two independent control volumes, steam and water. They are assumed to be in thermal equilibrium during steady-state condition. Turbine exhaust flow enters the system at the condenser pressure. The water part of the flow falls into the hotwell region and mixes perfectly with the water already present there. The vapor part condenses on the outer surface of metal tubes through which the circulating water flows. The condensation process is associated with a time delay which is due to the dynamics of heat transfer between the vapor and the circulating water. The following assumptions are made to simplify the condenser model.

- Constant pressure exists in the condenser model.
- Mass transfer at steam-liquid interface is assumed to be negligible.
- Wall condensation is considered to be zero.

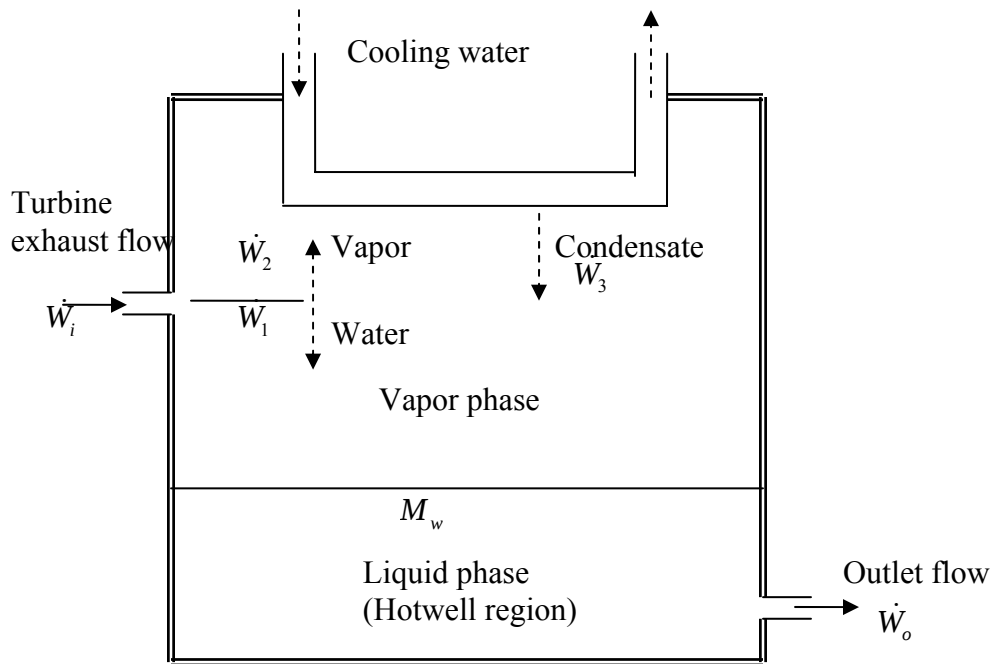


Figure 5.3. Schematic diagram of the turbine condenser model.

5.3.5 Feedwater Heater

A closed feedwater heater is basically a shell and tube heat exchanger where the feedwater flows through the tubes and interacts with extracted steam residing along the shell side. As the heat energy from the extraction steam is transferred to the feedwater, the steam condenses and collects at the bottom of the heater. Under normal conditions, the condensate from the heater is passed to the downstream lower pressure heater where it is used to increase the temperature of the feedwater. In emergency conditions, the condensate is directed through the emergency drain valve to the main condenser.

Most feedwater heaters used in power plants are of the shell and U-tube type, horizontal, three-zone configuration. A typical configuration is given in Figure 5.4. Feedwater entering the heater first passes through the drain-cooling region where single-phase convection is the leading heat transfer mechanism from the drains on the shell side to the feedwater on the tube side. The purpose of this section is to cool the drains to a temperature that is close to the shell side

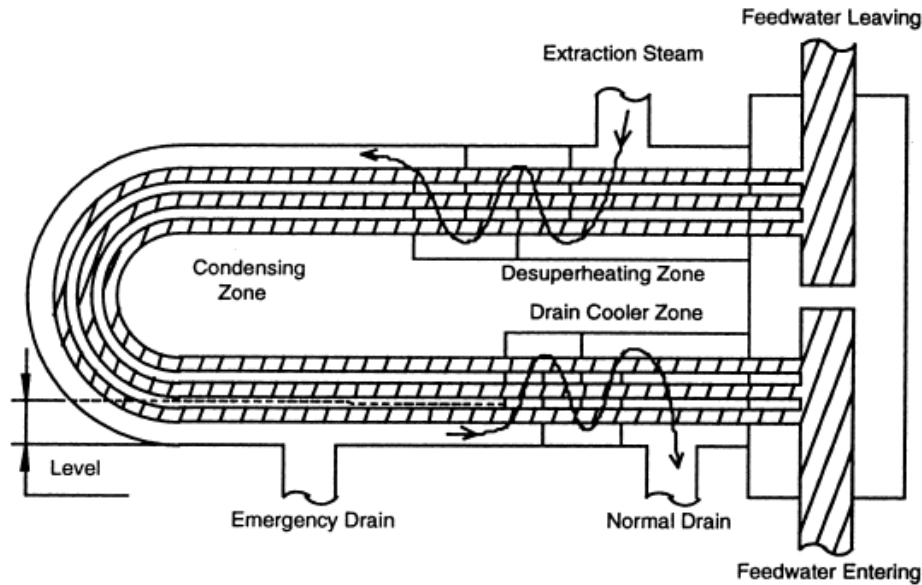


Figure 5.4. Schematic of a horizontal closed feedwater heater [Thanomsat et al., 1998].

saturation temperature of the next heater where it will be mixed with the extraction steam. The feedwater is then passed to the condensing region where the majority of the heat transfer takes place. The feedwater temperature can be brought up to within 5°C of shell side saturation temperature in this section. Finally, a de-superheating region is used to raise the feedwater temperature even above saturation and cool the steam down to saturation.

5.4 IRIS-MSF Nuclear Desalination Plant Simulation Results

Coupling an IRIS system with a MSF desalination plant requires co-location of the two units with a steam supply loop provided from the IRIS secondary system to the MSF plant. Electrical connection is also required to power the MSF pumps and auxiliary equipment. Coupled nuclear desalination plants, in fact, have been in operation for over two decades in countries such as Japan and Kazakhstan. Various concepts of the coupling of nuclear and desalination systems have been presented in the scientific community, some of which are already on their way to being demonstrated. An example of a possible nuclear MSF desalination plant complex is illustrated in Figure 5.5 [Ingersoll et al., 2004]. An intermediate steam loop is proposed for coupling two processes. The pressure of the intermediate loop is maintained above

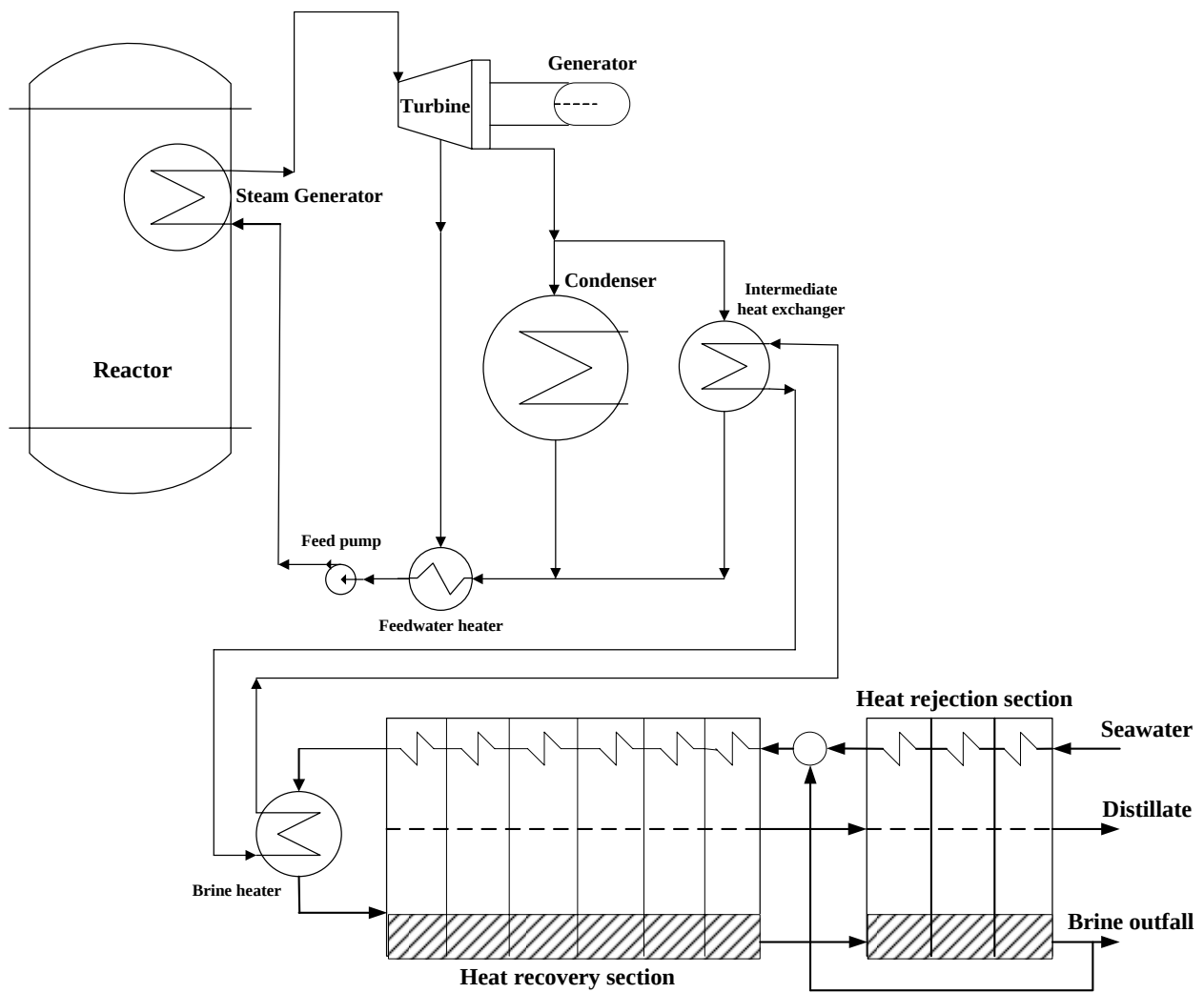


Figure 5.5. Schematic diagram of a MSF plant coupled to the IRIS power conversion system.

the pressure of the secondary IRIS loop and the pressure of the seawater loop in the brine heater stage of the MSF plant so that a failure of the heat exchanger at either end of the intermediate loop would result in an easily detectable pressure drop in the loop and ensure isolation of the fluids in the two processes.

In this dissertation, the MSF brine heater steam supply is obtained directly from the exhaust of the low-pressure steam turbine. The heating steam has an initial saturation temperature of $206^{\circ}F$. The brine flows through the tubing in the heat exchanger. The condensate gathered in the sump tank of the brine heater mixes with the flow from the main condenser, and goes to the feedwater heaters.

The overall simulation model of the coupled IRIS-MSF plant is developed by combining the individual SIMULINK models of reactor core, helical coil steam generators, balance-of-plant, as well as the MSF system. The order of the coupled IRIS-MSF SIMULINK model is estimated around more than 100 state variables.

The simulation results of the typical variables in the coupled system due to a 10% step decrease in the power demand (from 100% to 90%) are shown in Figures 5.6 - 5.18.

Figure 5.6 shows the reactor core power change due to the power demand decrease. The core power, which is initially around 100%, goes to approximately 93.5% in 400 seconds. Figures 5.7 - 5.9 illustrate the responses of T_{hot} , T_{cold} , and T_{ave} . As the core power decreases, T_{hot} goes down as well. The T_{ave} controller adjusts the external reactivity to maintain a constant average coolant temperature around the set point ($590^{\circ}F$). The steam outlet pressure response shown in Figure 5.10 also varies due to the power change, but the steam pressure controller is able to keep the pressure as close as possible to the set point (841 PSI) by reducing the steam flow rate to the turbine system, shown in Figure 5.11. This, in turn, results in a reduction in the turbine power, as shown in Figure 5.12. It also causes a reduction in the heat flow to the feedwater heater that consequently decreases the feedwater temperature (Figure 5.13). As the power varies, so does the steam generator water level. Hence more heat transfer area is left for the steam to be superheated, thus increasing the steam temperature (Figure 5.14).

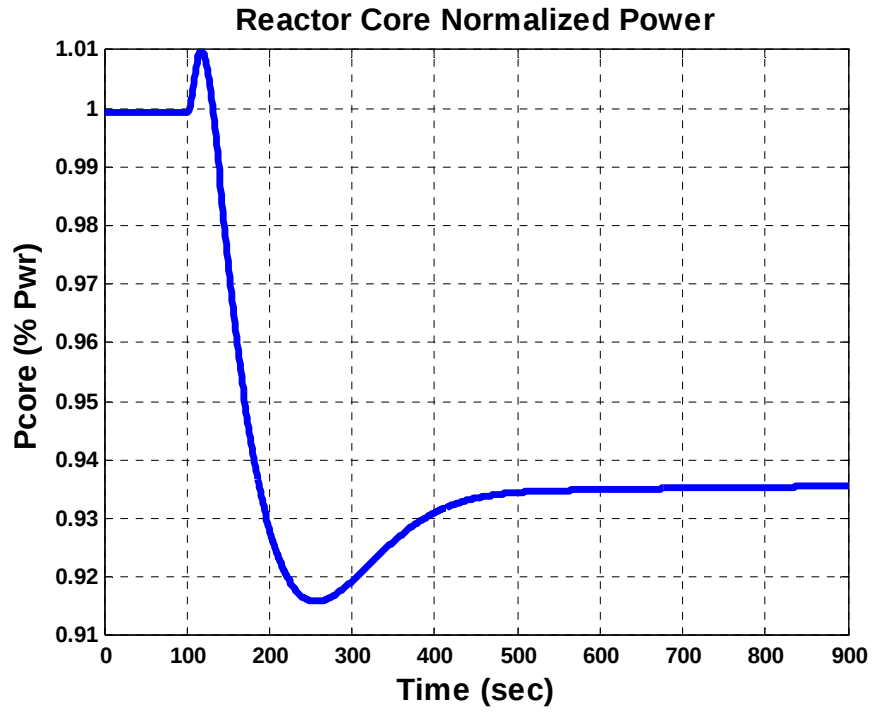


Figure 5.6. Reactor power response to a 10% power demand step decrease.

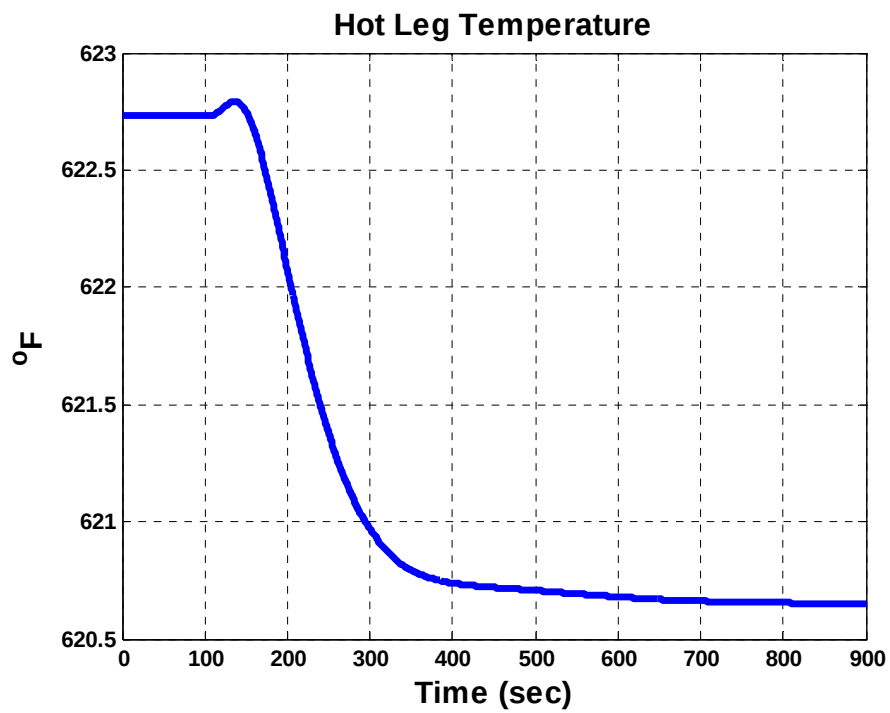


Figure 5.7. T_{hot} response to a 10% power demand step decrease.

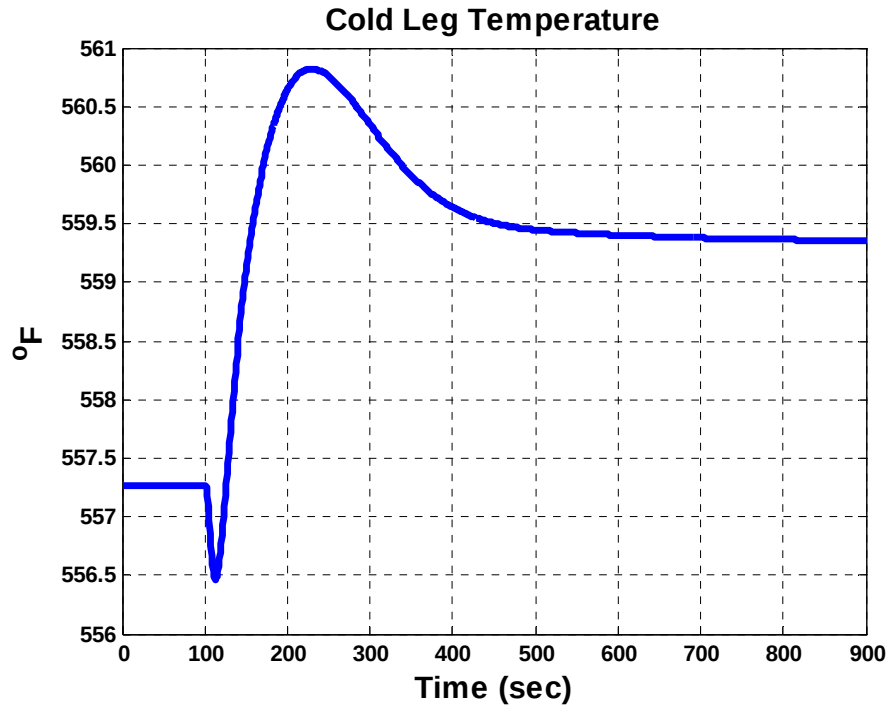


Figure 5.8. T_{cold} response to a 10% power demand step decrease.

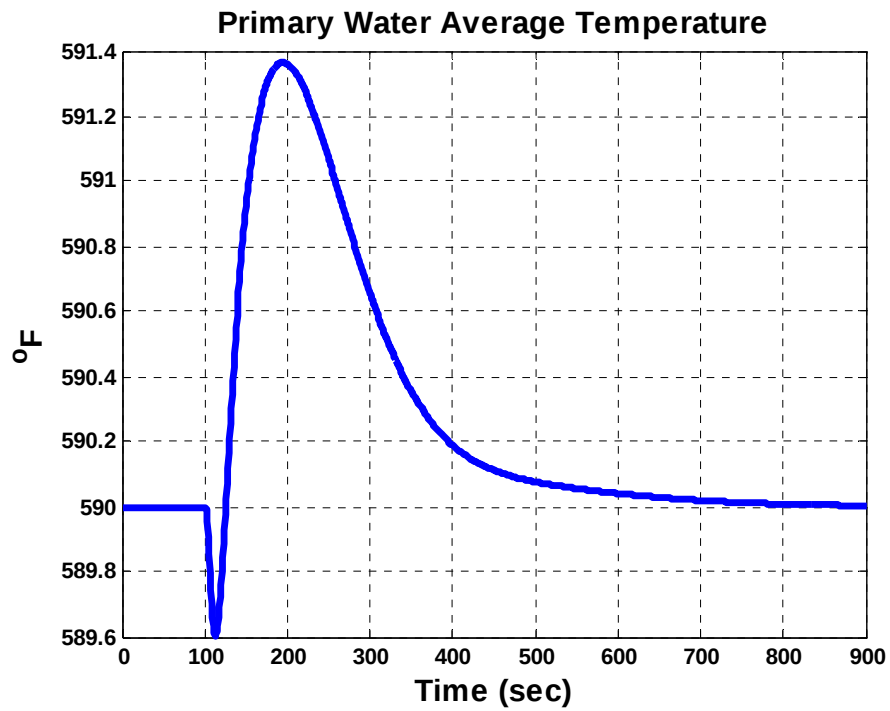


Figure 5.9. T_{ave} response to a 10% power demand step decrease.

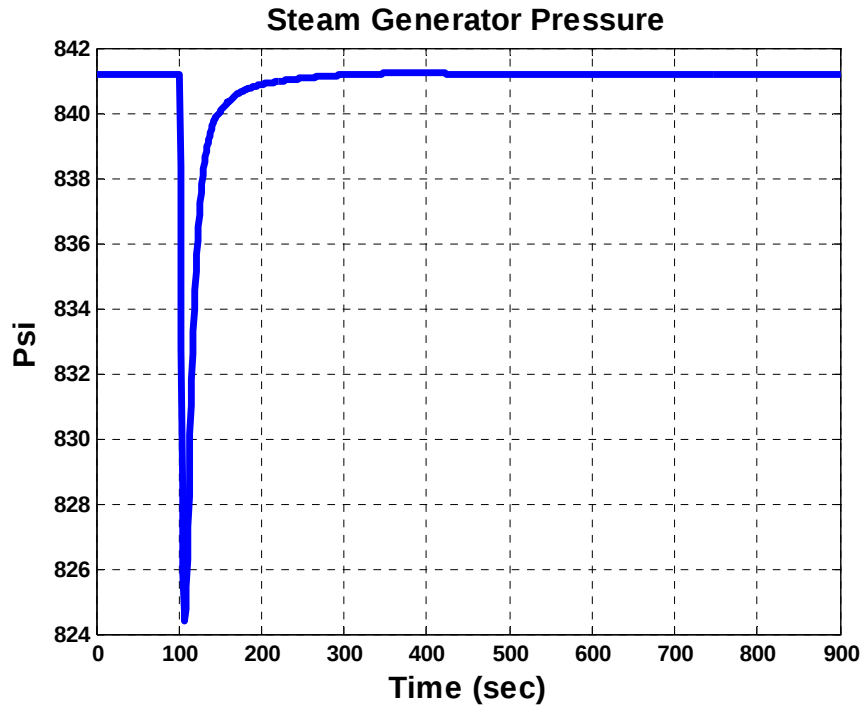


Figure 5.10. Steam pressure response to a 10% power demand step decrease.

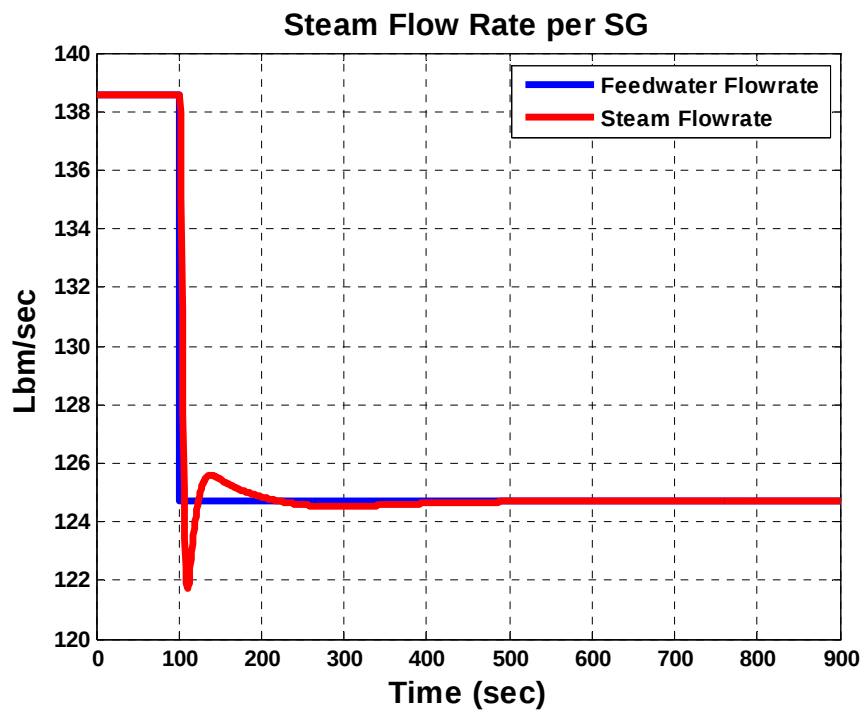


Figure 5.11. Steam flow rate response to a 10% power demand step decrease.

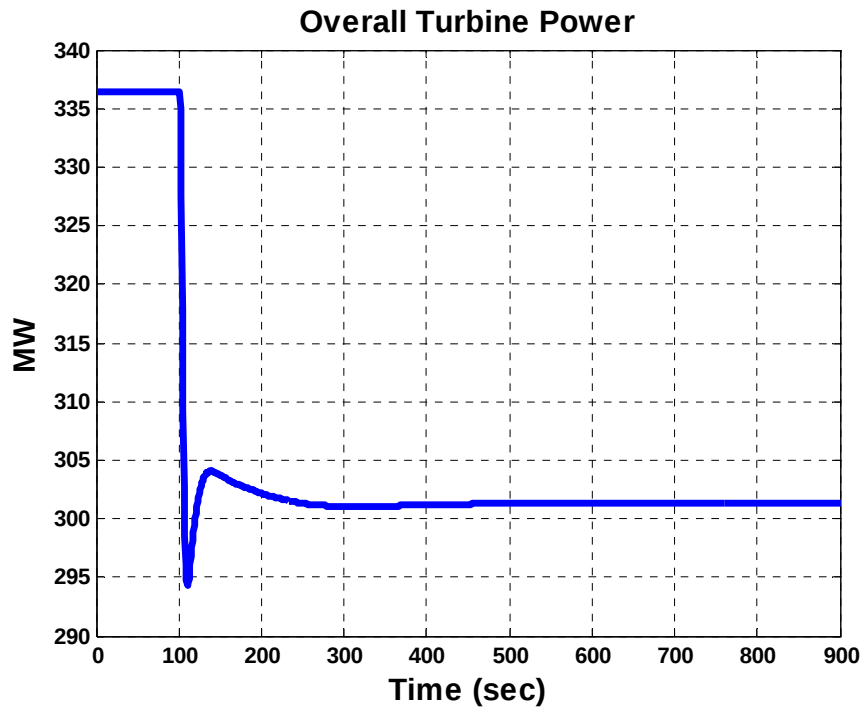


Figure 5.12. Turbine output response to a 10% power demand step decrease.

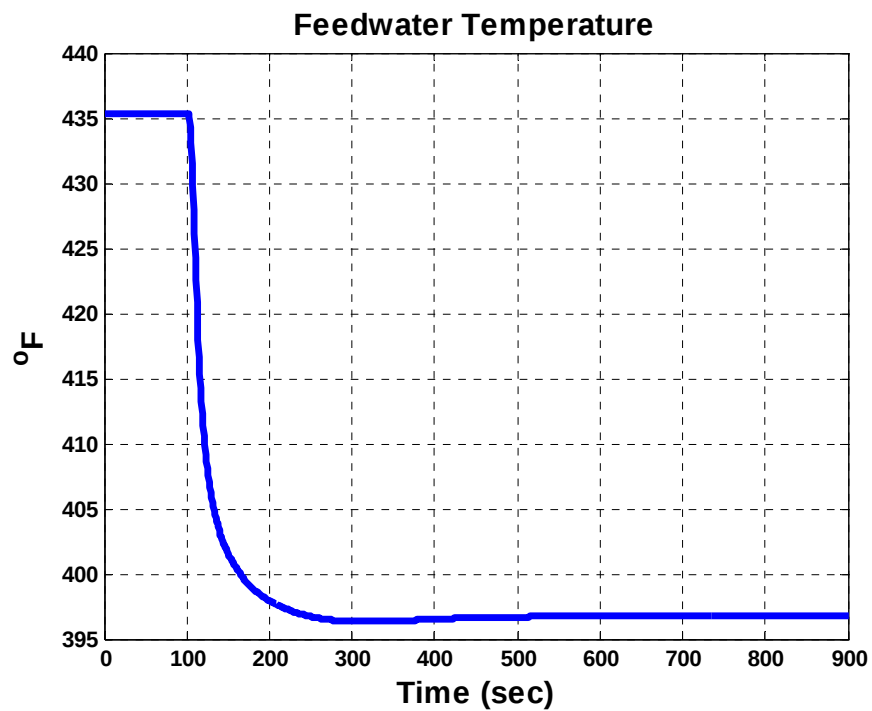


Figure 5.13. Feedwater temperature response to a 10% power demand step decrease.

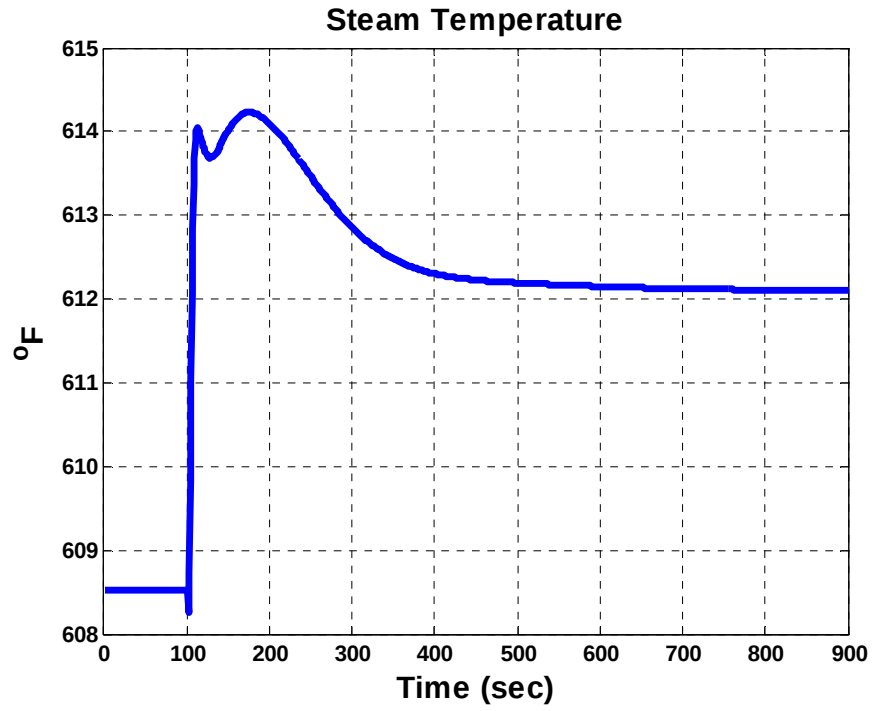


Figure 5.14. Steam temperature response to a 10% power demand step decrease.

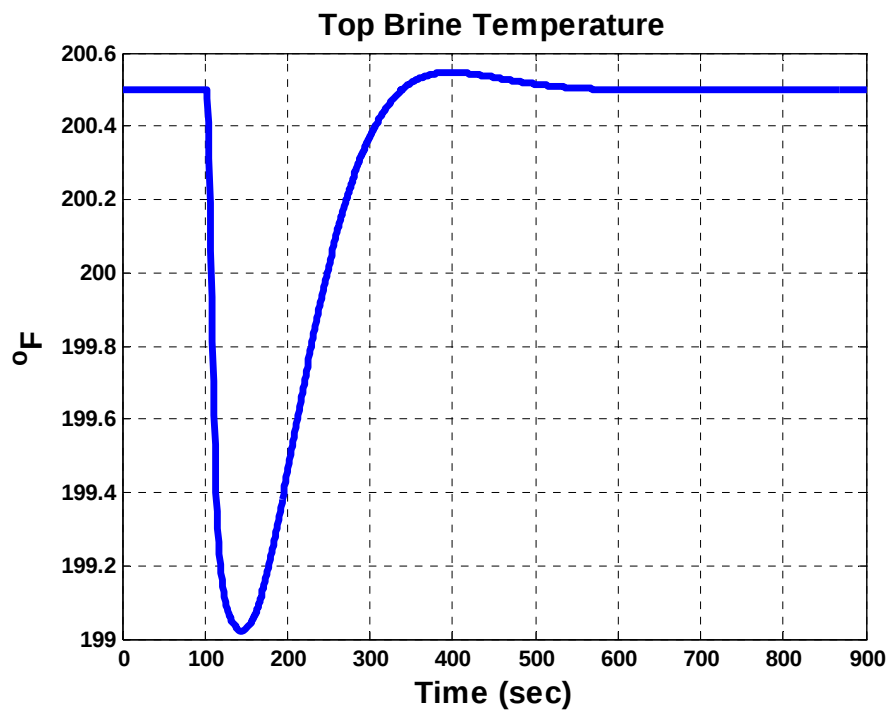


Figure 5.15. Top brine temperature response to a 10% power demand step decrease.

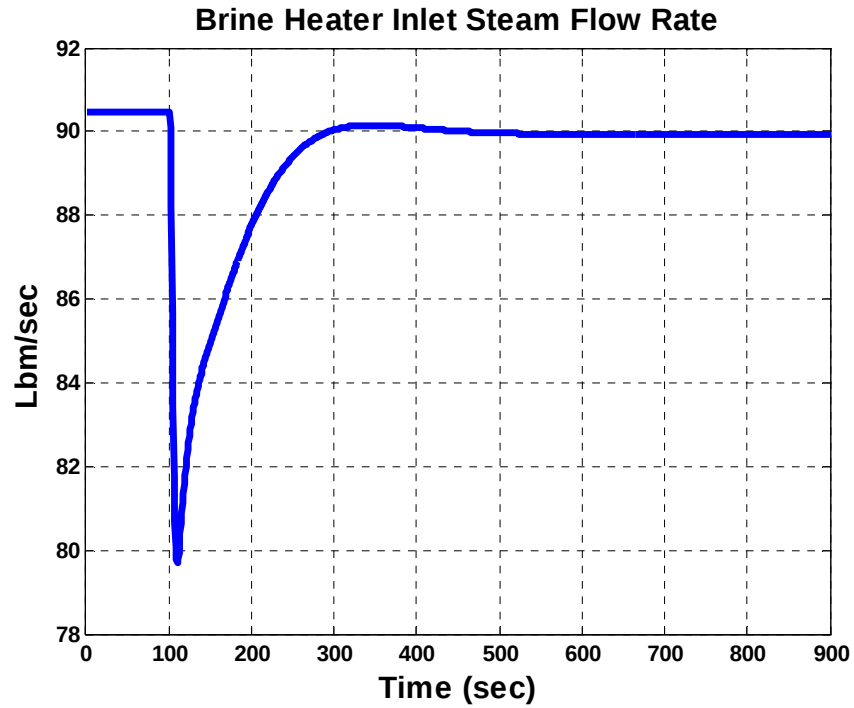


Figure 5.16. Brine heater inlet steam flow rate response to a 10% power demand step decrease.

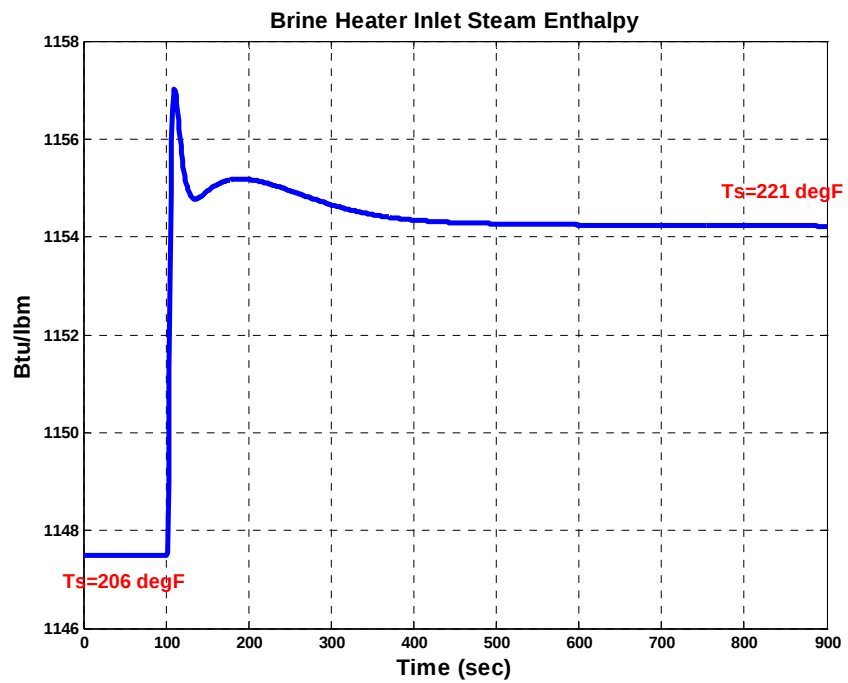


Figure 5.17. Brine heater inlet steam enthalpy response to a 10% power demand step decrease.

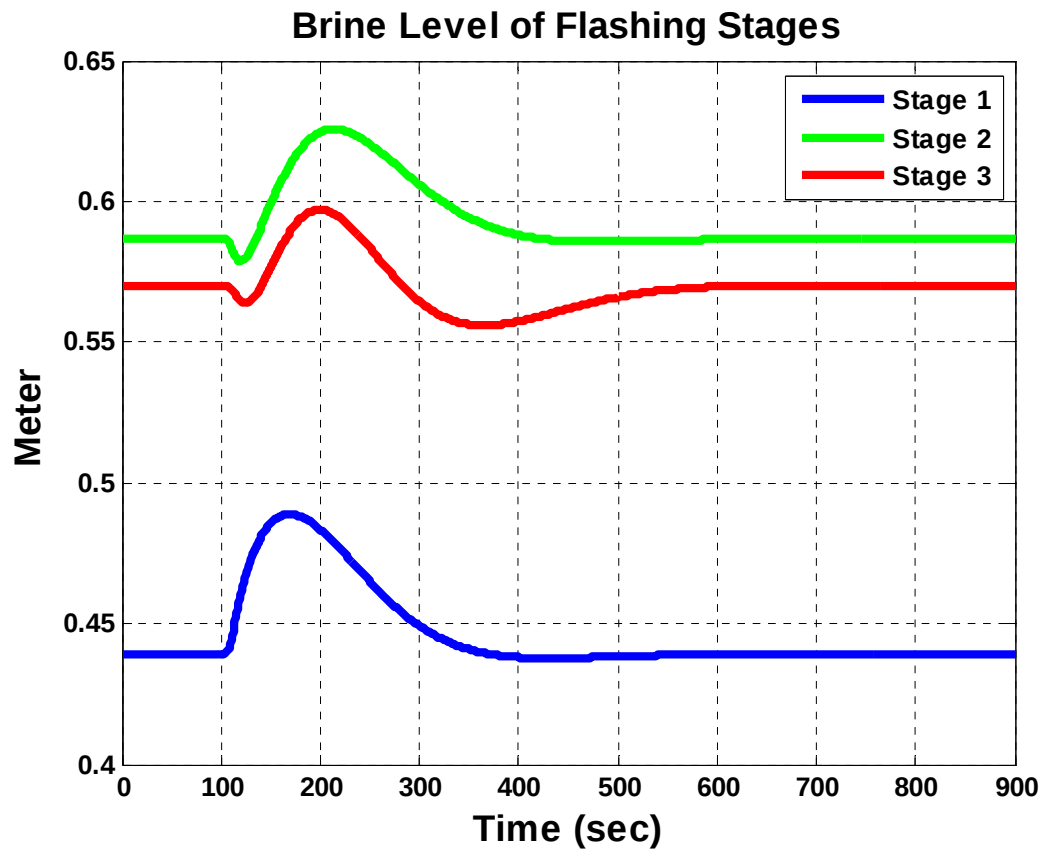


Figure 5.18. Brine level response to a 10% power demand step decrease.

5.5 Summary

In this chapter, the development of the coupled IRIS-MSF nuclear desalination system model has been described. An existing IRIS plant SIMULINK model was used for coupling with the dynamic MSF model developed and described in Chapter 4. A simple BOP model was also described in this chapter. The BOP model incorporated the subsystem models of the turbine, condenser, and feedwater heater systems. The dynamic model of the coupled plant has been simulated for a 10% step decrease in the turbine power demand, and the simulation results have been analyzed in this chapter. The results showed that the developed dynamic model was able to characterize the system dynamic behavior with reasonably good accuracy to study the control and fault diagnosis of the IRIS-MSF coupled system.

Chapter 6

Development of Sensor Placement Design Algorithms

6.1 Introduction

Sensor allocation has been treated as an optimization problem in most of the earlier work. The first attempt to present a technique to locate sensors was made by Lambert [1977], where he used probabilistic importance of events in fault trees to decide optimal sensor locations. Vaclavek and Loucka [1976] described the problem of sensor network design and employed graph theory to ensure the observability of a specified set of important variables in a multi-component flow network. Ali and Narasimhan [1993] addressed the concept of reliability of state variable estimation and developed graph-theoretic algorithms for maximizing the reliability. The reliability of the process was defined as the smallest reliability among all of the variables. Unlike the approaches based on graph theory and linear algebra, Bagajewicz [1997] proposed a mixed integer nonlinear programming (MINLP) problem to obtain cost-optimal structures subject to the desired level of precision, residual precision, and error detectability. An alternative mixed integer linear programming (MILP) formulation which was useful for both small- and medium-size problems was presented by Bagajewicz and Cabrera [2002]. Sen et al. [1998] presented a genetic algorithm based approach that can be applied for the design of non-redundant sensor networks using different objective functions.

In this dissertation, the graph-based techniques are used to optimize sensor locations. Issues such as fault detectability and discriminability are discussed to ensure the designed sensor network could observe every defined fault in the process, meanwhile obtain a maximum possible fault resolution. Directed graph (DG) that represents cause-effect behavior of the process is used as a basis for the sensor placement design. DG is employed to describe the propagation of fault effects, and a cause-effect analysis is performed based on the graph theory to design the sensor placement strategy.

The contribution of this dissertation to the field of optimum sensor placement design is innovative in that it develops an optimal and automated sensor allocation procedure for an IRIS-MSF nuclear desalination plant, with considerations of both single-fault and dual-fault assumptions in the application. In addition, this dissertation proposes an efficient integer linear programming (ILP) embedded greedy search algorithm to solve the formulated sensor allocation optimization problem. And the optimal results are compared with those from LINGO 8.0, a commercial ILP optimization solver.

6.2 Methodology of Directed Graph Modeling

The solution to the problem of sensor placement may be broadly broken down into two tasks: (1) fault modeling or prediction of cause-effect behavior of the system, generating a set of variables that are affected whenever a fault occurs, and (2) use of the generated sets to identify sensor locations based on various design criteria, such as fault observability, fault resolution, etc. The fault propagation or cause-effect behavior is derived on the basis of a qualitative model that is used to represent the process.

Directed graph or digraph is such a qualitative cause-effect model that can be used to infer the mechanism of fault propagation in a system. It normally consists of a set of nodes and directed branches. The nodes in a DG model represent states of process variables or malfunctions, and the branches correspond to the causal influences between the nodes. The arrow of a branch reflects the direction of the effect between variables, and the fault propagation pattern can thus be inferred graphically. Based on the DG model of a given process, approaches from the graph theory are then employed to design the sensor networks according to some specified criteria.

The DG modeling is a convenient approach because it clearly illustrates the interactions among the important process variables, and can be easily developed from the empirical relationships or engineering fundamental principles. Figure 6.1 shows a DG diagram of a hypothetical process with the path connections from faults to potential location of sensors, as an illustrating example. In this figure, each node corresponding to a fault (F_j) in the process connects through an arrow to a sensor (S_i), thus indicating that the fault will affect the reading of

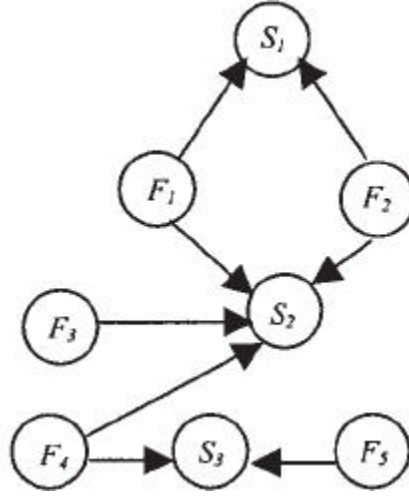


Figure 6.1. Directed graph of a hypothetical process.

the corresponding sensor. The method used to solve the problem of sensor location is based on identifying directed paths from root nodes where faults can occur to nodes where effects can be measured, called the observability set. Using all these paths, the objective is to choose the minimum subset of sensors from the observability set that would have at least one directed path from every root node.

The fault-and-sensor maximum connectivity matrix D is also generated for the hypothetical process. The rows of this matrix represent the sensors, and the columns represent the faults. The $(i, j)_{th}$ entry D_{ij} of this matrix is 1 if fault F_j affects the potential sensor location S_i and is 0 otherwise ($i = 1, 2, \dots, M$ and $j = 1, 2, \dots, N$), where M is the total number of sensors and N is the total number of faults. The generated matrix is given by:

$D =$

	F_1	F_2	F_3	F_4	F_5
S_1	1	1	0	0	0
S_2	1	1	1	1	0
S_3	0	0	0	1	1

6.3 Sensor Location and Fault Diagnostic Observability Criterion

6.3.1 Set-Covering Problems

The sensor placement design that makes use of optimization criteria of fault observability and fault discriminability considers finding the optimal sensor locations as one of choosing the minimum number of sensors that would cover all the faults (root nodes). This is the well-known “minimum set-covering problem”. The greedy search algorithm developed by Raghuraj et al. [1999], despite one of the fastest solving methods, does not necessarily find all minimal sensor sets [Krysander et al., 2008]. For this purpose, a literature review on solving set-covering problems using heuristic algorithms has been performed.

Set-covering problem is a classical problem in computer science and complexity theory. Set-covering problem is one of the most important discrete optimization problems because it serves as a model for real world problems. Real world problems that can be modeled as set-covering problems include facility location problem, airline crew scheduling, nurse scheduling problem, resource allocation, assembly line balancing, vehicle routing, etc. Set-covering problem is a problem of covering the rows of an $m \times n$ zero-one matrix with a subset of columns at minimal cost [Beasley and Chu, 1996]. Set-covering problem can be formulated as follows:

$$\text{minimize } [\sum_{j=1}^n c_j x_j] \quad (6.1)$$

subject to

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m \quad (6.2)$$

$$x_j \in \{0, 1\}, \quad j = 1, \dots, n \quad (6.3)$$

Equation (6.1) is the objective function of set-covering problem, where c_j ($j = 1, \dots, n$) is refer to weight or cost of covering column j and x_j is decision variable. Equation (6.2) is a constraint to ensure that each row is covered by at least one column where a_{ij} is constraint coefficient matrix of size $m \times n$ whose elements comprise of either “1” or “0”. A column j

($j=1,\dots,n$) is said to cover a row i ($i=1,\dots,m$) if $a_{ij}=1$. Set-covering problem calls for a minimum cost subset S , such that each row i ($i=1,\dots,m$) is covered by at least one column, $j \in S$. Finally, Equation (6.3) is the integrality constraint in which the value is represented as in Equation (6.4).

$$x_j = \begin{cases} 1 & \text{if } j \in S, \\ 0 & \text{otherwise.} \end{cases} \quad (6.4)$$

Even though it may seem to be a simple problem by judging from the objective functions and constraints of the problem, set-covering problem is a combinational optimization problem.

A number of heuristic algorithms for set-covering problem have been reported in the literature. Beasley, as one of main researcher in set-covering problem had implemented several algorithms in order to solve set-covering problem. Beasley presented an algorithm that combines problem reduction tests with dual ascent, sub-gradient optimization and linear programming. This algorithm had performed well in solving set-covering problem [Beasley, 1987]. It was able to find feasible optimal solutions for all set-covering problem sets. In a different literature, Beasley presented a paper which used Lagrangian relaxation and sub-gradient optimization approach to solve the set-covering problem [Beasley, 1990]. But this method did not perform well compared to his previous method. It was unable to find optimal solutions for several set-covering problems. Haddadi presented a simple Lagrangian heuristic to solve set-covering problem [Haddadi, 1997]. The method is based on Lagrangian duality, greedy heuristic for set-covering problem, sub-gradient optimization and redundant covers. This method had turned out to be efficient for low density set-covering problems with a large number of variables with average deviation of 0.35%.

Beasley and Chu used genetic algorithm for set-covering problem [Beasley and Chu, 1996]. They presented a new crossover-fusion operator, a variable mutation rate and a heuristic feasibility operator to improve the performance of genetic algorithm. This method performs well in solving most problems. Aickelin proposed an indirect genetic algorithm [Aickelin, 2002]. The indirect genetic algorithm comprises of three phases. In the first phase, the genetic algorithm finds good permutation of the rows to be covered. In the second phase, a decoder builds a

solution from the permutations using the parameter provided. And lastly, in the third phase, a hill-climber optimization method is used. The indirect genetic algorithm is able to solve the set-covering problem in a shorter computational time. Monfreglio proposed a linear programming relaxation model and improvement techniques based on simulated neural network [Monfreglio, 1998]. This method is able to find solutions within 0.2% of optimal solution and increase the overall computational time. Vasko and Wolf adapted heuristic concentration approach to solve the weighted (non-unicost) set-covering problem [Vasko and Wolf, 2001]. Their method is able to solve set-covering problem and find solution with deviation of maximum of 3.27% from optimum solution.

6.3.2 Fault Observability

Fault observability refers to the condition that every fault defined for the process has to be observed by at least one sensor. Given a process DG model, the observability problem becomes one of finding the minimum number of sensors that would cover all the faults in the process (root nodes). This is commonly known as “minimum set-covering problem” [Parker and Rardin, 1988], where the sets to be covered are the sets of sensors affected by each process fault. Although a simple enumeration can be made for a small system, a systematic approach needs to be used for a large system. In this approach, the first step is to build a bipartite graph, which consists of a causal set including all the fault nodes and an observability set including all the sensor nodes with only input arcs in the DG model. A graph is bipartite if a vertex set can be partitioned into two sets in such a way that no two vertices from the same set are adjacent. These two sets constitute a bipartition of the original vertex set [Asratian et al., 1998]. After a bipartite graph is obtained from the DG model, a subset of the key variables can be chosen from the observability set as the minimum sensor requirement for fault detection based on a greedy search algorithm developed by Raghuraj et al.[1999].

This search algorithm is summarized as follows:

- (1) Initiate C and G as empty sets.

(2) Construct a bipartite graph between the fault nodes and the key variables in the fault observability set.

(3) Select a variable among the unmarked key variables based on the largest number of incident arcs.

(4) Mark the selected key variable in step (2) and store it in C .

(5) Determine all the fault nodes covered by C .

(6) If there exist uncovered fault nodes, delete all the arcs from the fault nodes covered by the selected variable to all the previously marked key variables; store in a buffer set G all the arcs from the fault nodes covered by the selected variable to the unmarked key variables, then go back to step (3); else, remove the variables from C that do not have arcs incident on them based on the arcs stored in G . Output the set C and stop.

(7) End.

The number of incident arcs used in step (3) is defined as the difference between the actual number of arcs incident on a key variable and the number of arcs incident on the same key variable that have been stored in the buffer set G . After the algorithm is completed, the same set of key variables can be obtained in C as is obtained from the greedy search algorithm. However, the redundant key variables will be stored by tracking the arcs stored in the buffer set G that do not have a connection with the marked key variables. To determine the minimum set of sensors, the redundant key variables must be removed from the obtained key variables stored in C .

Example:

Let us use the hypothetical system shown in Figure 6.1 as a simple example to illustrate the greedy search algorithm based on fault observability criteria. In this example, there are five fault classes F_1, F_2, F_3, F_4 , and F_5 , and three key variables S_1, S_2 , and S_3 on which sensors can be placed. Figure 6.2 depicts the bipartite graph for the original digraph.

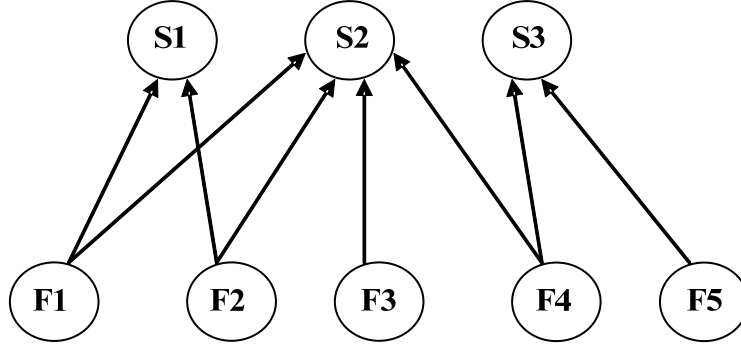


Figure 6.2. Fault and sensor node bipartite graph for Figure 6.1.

Figure 6.2 clearly shows that the set $[S_1, S_2, S_3]$ form the “observability set”, as these are the nodes in the DG model with only input arcs. It is also clear that by placing sensors on these nodes, one could detect all the faults. However, the set $[S_1, S_2, S_3]$ is not the minimum set. The minimum set is, in fact, $[S_2, S_3]$. As a first step in the algorithm, node S_2 is chosen as the key component, as it has the maximum number of arcs incident on it. In the mean time, the arcs from F_1 to S_1 , F_2 to S_1 , and F_4 to S_3 will be stored in set G because F_1, F_2 , and F_4 are covered by the selected variable S_2 , which is marked. In the next step, S_3 is chosen as the next key component, and with this it is clearly seen that all the fault nodes are now covered. Both S_2 and S_3 have the arcs stored in set G incident on it. This is the minimum observability set, as given by the greedy search algorithm.

6.3.3 Fault Resolution

Fault resolution refers to the ability to identify the exact fault within a system of interest that has occurred. The maximum resolution that can be attained is restricted by the topology of DG and the position of the fault or root nodes in a DG model. Hence, given the constraints on measurement points, the problem of fault resolution is equivalent to generating sensor locations so that every fault is resolved to the maximum extent possible. This condition is referred to as the “highest fault resolution”. As shown by Raghuraj et al. [1999], any fault resolution problem (single-fault resolution, multiple-fault resolution, etc.) can be converted to a suitable fault

observability problem. Also, the assumption of single-fault or multiple-fault would lead to different fault resolution solutions.

Single-Fault Resolution

For each pair of faults i and j , the fault set $B_{ij} = A_i \cup A_j - A_i \cap A_j$ is generated. In this expression, A_i and A_j are the sets of measurable nodes connected to the i_{th} fault and j_{th} fault, respectively. Note that set B_{ij} represents the symmetric difference of sets A_i and A_j , which consists of the set of nodes that can be used for differentiating between a pair of fault nodes i and j . Thus, each B_{ij} is treated as a virtual fault, and a bipartite graph is constructed between these nodes and the sensor nodes. This bipartite graph is then added to the original bipartite graph.

The minimum set of sensors for fault resolution can also be determined using the greedy search algorithm. The steps are summarized as follows:

- (1) Determine the set of key variables consisting of the different members of A_i and A_j that covers the virtual fault B_{ij} , which denotes the resolution of the original fault nodes i and j .
- (2) Construct a bipartite graph between the set of virtual fault nodes and their observability set.
- (3) This new bipartite graph is added to the original bipartite graph, which is between root nodes and sensor nodes.
- (4) Apply the greedy search algorithm developed for fault observability to the updated bipartite graph to determine the minimum set of sensor nodes that would cover all the root nodes and the virtual fault nodes. The selected sensor set based on this algorithm can ensure that the specified faults can be detected and distinguished.

Example:

To illustrate the concepts developed, fault diagnosis of the previous example is performed. The greedy search algorithm is used for fault diagnosis under the single-fault assumption. As a first step, the virtual faults $B_{ij} = A_i \cup A_j - A_i \cap A_j$ are constructed for all i and j . The sensor sets for the different faults are given in Table 6.1. Some faults may affect the identical sensor set. In this case, they have been combined as one fault, and cannot be distinguished from one another. For example, nodes F_1 and F_2 share the same sensor set $[S_1, S_2]$, and have been combined and denoted by the node A_1 . The set B_{ij} for the faults are shown in Table 6.2. The faults consist of the original faults plus the virtual fault set B_{ij} . The greedy search algorithm is applied on this extended fault system and the optimum sensor set is found to be $[S_1, S_2, S_3]$. All the original faults in Figure 6.2 can be distinguished from one another by the chosen sensor set, except for F_1 and F_2 .

Multiple-Fault Resolution

Assuming multiple faults, the sensor placement problem is solved as an extension of the single-fault resolution problem. For the sake of illustration, consider the specific case when a maximum of two faults can occur at a time. For each pair of faults i and j , the set $A_{ij} = A_i \cup A_j$ is formed, where A_{ij} represents the set of nodes which are affected when both of the faults i and j occur simultaneously. The set A_{ij} is treated as a virtual fault, and is therefore added to the original set of faults.

Solving the single-fault resolution problem for this new system gives the sensor locations for resolution under dual-fault assumption. The same methodology can be applied to cases where more than two faults occur simultaneously. Clearly, with a greater number of sets generated, the computational complexity of the approach increases, but as discussed in Raghuraj et al. [1999], one might not be interested in all multiple-fault situations and the above formulation gives the designer a methodology by which different sensor location problems can be posed and solved.

Table 6.1. Set A of sensors for different fault nodes in Figure 6.2

Fault Nodes	Sensor Nodes	Set A
F_1, F_2	$[S_1, S_2]$	A_1
F_3	$[S_2]$	A_2
F_4	$[S_2, S_3]$	A_3
F_5	$[S_3]$	A_4

Table 6.2. Set B_{ij} for single-fault case of Table 6.1

Set B	Sensor Nodes
B_{12}	$[S_1]$
B_{13}	$[S_1, S_3]$
B_{14}	$[S_1, S_2, S_3]$
B_{23}	$[S_3]$
B_{24}	$[S_2, S_3]$
B_{34}	$[S_2]$

6.4 Sensor Location and System Unobservability Criterion

6.4.1 Formulation of Unobservability Minimization

The design of a reliable instrumentation system for an industrial process is of the ultimate importance. A sensor network is highly reliable if the probability of any fault occurring without being detected is low. For a given process, the faults in that process have certain occurrence probabilities. The various available sensors also have certain failure probabilities, which depend on the type of sensor and the variable being measured. The only way in which a fault can occur without being detected is that the fault occurs and the sensors monitoring that fault fail simultaneously. The probability of such an event taking place is the product of the fault occurrence probability f_j and the corresponding sensor failure probability s_i , which is defined as the *unobservability value* U_j of the j_{th} fault.

$$U_j = f_j \prod_{i=1}^M s_i^{D_{ij}x_i}, j = 1, 2, \dots, N \quad (6.5)$$

In Equation (6.5), x_i is the number of sensors being placed at a particular location, and the binary number D_{ij} is the $(i, j)_{th}$ entry of the fault-sensor maximum connectivity matrix. Note that in the formulated optimization problem, x_i is allowed to be greater than 1, which, in other words, means hardware redundancy is allowed. This feature makes the approach practical because it is reasonable to use more than one sensor to measure a variable if that particular sensor node has a high tendency to fail or if the covered fault has a high probability of occurrence.

Another useful quantity (*redundancy value*) used in the algorithm is defined in Equation (6.6). This value provides an approximate measure of the total unobservability covered by a sensor.

$$R_i = \sum_{j=1}^N D_{ij}U_j, i = 1, 2, \dots, M \quad (6.6)$$

Thus, achieving a minimum unobservability of the system is paramount for the design of a reliable sensor network. The sensor placement formulation is based on minimizing the system overall unobservability, which is defined as the maximum unobservability among all faults ($\text{Max}_{\forall_j} U_j$). This concept is based on the philosophy that a chain can be no stronger than its weakest link. Therefore, the optimization of sensor placement design is formulated as follows:

$$\underset{x_i}{\text{minimize}} \left[\text{Max}_{\forall_j} U_j \right] \quad (6.7)$$

subject to

$$\sum_{i=1}^M c_i x_i \leq TC \quad (6.8)$$

$$\sum_{i=1}^M D_{ij}x_i \geq 1, j = 1, 2, \dots, N \quad (6.9)$$

$$x_i \in \mathbb{Z}^+ \cup \{0\}, i = 1, 2, \dots, M \quad (6.10)$$

Equation (6.7) is the objective function of the formulated unobservability optimization problem, where U_j is given in Equation (6.5). The constraint given by Equation (6.8) ensures the utilized cost does not exceed the total available cost TC , where c_i is the cost of placing a sensor at node i and x_i is the decision variable. Equation (6.9) is an observability constraint to ensure that each column (fault) is covered by at least one row (sensor location) where D_{ij} is the $(i, j)_{th}$ entry of the fault-sensor maximum connectivity matrix whose elements comprise of either '1' or '0'. Finally, Equation (6.10) allows the decision variables x_i to take nonnegative integer values which may be greater than 1. For instance, $x_i = 4$ indicates that four sensors are placed at the same location.

Even though it may seem to be a simple problem by judging from the objective functions and constraints of the problem, the above problem is an integer nonlinear programming (INLP) problem which is not easy to solve exactly. However, the problem can be converted to a linear integer programming problem through a proper transformation.

The objective function in Equation (6.7) can be replaced by a linear objective function:

$$\underset{x_i}{\text{minimize}} \left[\underset{\forall_j}{\text{Max}} \ln(U_j) \right] \quad (6.11)$$

where

$$\ln(U_j) = \ln(f_j) + \sum_{i=1}^M D_{ij} x_i \ln(s_i), \quad j = 1, 2, \dots, N \quad (6.12)$$

Thus, $\ln(U_j)$ is linear in the decision variables x_i and is obtained by taking the natural logarithm on both sides of Equation (6.5).

The objective function given in Equation (6.11) is still not in the standard ILP form because it involves minimization of the maximum value. Therefore, it is being modified again.

$$\underset{x_i}{\text{minimize}} [U] \quad (6.13)$$

subject to

$$U \geq \ln(U_j), j = 1, 2, \dots, N \quad (6.14)$$

$$\sum_{i=1}^M c_i x_i \leq TC \quad (6.15)$$

$$\sum_{i=1}^M D_{ij} x_i \geq 1, j = 1, 2, \dots, N \quad (6.16)$$

$$x_i \in \mathbb{Z}^+ \cup \{0\}, i = 1, 2, \dots, M \quad (6.17)$$

The objective function given in Equation (6.13) is now in the standard ILP form.

6.4.2 One-Step Optimization Formulation

It has been found that the above optimization problem typically has multiple solutions for unobservability of detection with different costs. Lexicographic optimization [Bhushan and Rengaswamy, 2000a] is one way of finding the solution with the least cost among these multiple solutions. This is done by obtaining a solution that optimizes the objectives in an ordered fashion. Minimizing the unobservability is the first objective and minimizing the cost is the next objective. Such an ordering would mean that a higher level objective is more important than a lower level objective. We will show below a single objective function that combines system unobservability minimization and cost minimization.

$$\underset{x_i}{\text{minimize}} [U - \alpha x_s] \quad (6.18)$$

subject to

$$U \geq \ln(U_j), j = 1, 2, \dots, N \quad (6.19)$$

$$\sum_{i=1}^M c_i x_i + x_s \leq TC \quad (6.20)$$

$$\sum_{i=1}^M D_{ij} x_i \geq 1, j = 1, 2, \dots, N \quad (6.21)$$

$$x_i \in \mathbb{Z}^+ \cup \{0\}, i = 1, 2, \dots, M \quad (6.22)$$

$$x_s \in \mathbb{R}^+ \cup \{0\} \quad (6.23)$$

In the above problem, the objective function U is the maximum unobservability of detection (on a natural log scale) among all faults, which is ensured by constraints in Equation (6.19). The variable x_s is the slack in the cost constraint, which takes nonnegative real values.

The higher the value of x_s , the less cost is used for sensor placement. α is a positive constant which has to be selected such that the primary objective (minimizing unobservability) still attains its earlier optimal value. Among all solutions which yield minimum system unobservability, the one which has the highest x_s will be chosen. Therefore, if the constant α is appropriately selected, the solution will give a sensor network that has the least cost among all the potential sensor networks that yield the minimum system unobservability.

6.4.3 Solving Unobservability Minimization Problems

The problems presented above can be solved by standard solvers for moderate size problems. In this dissertation, the commercial ILP optimization software LINGO 8.0, which employs a branch and bound enumeration algorithm, is used to solve the formulated unobservability minimization problem. Also, a greedy search heuristic is developed to find the optimal number of sensors and their placement. The detailed procedure of the algorithm is described below. At each step the best available sensor is chosen. Also, a flowchart of the algorithm is illustrated in Figure 6.3.

Step 1: Initialize the following variables: current solution (PS), optimal solution (OS), total available cost (TC), cost currently utilized (C), optimal cost utilized (OC).

Step 2: The fault with the highest unobservability value is selected. In the case of more than one fault having the highest unobservability, any one among these faults may be selected.

Step 3: Select a sensor which observes the fault chosen in the previous step, and try to lower the unobservability value of that fault. However, there may be more than one qualified sensor. Among these sensors, only those whose cost is less than the currently available cost, $TC-C$, can be chosen. Let S be the set of sensors which are connected to the selected fault, as well as cost of each member of S is less than or equal to $TC-C$. A sensor of S with the minimum failure probability is to be chosen because this causes a maximum decrease in the unobservability value of the fault under consideration. If more than one element of S satisfies this criteria (that is more than one element of S has the lowest failure probability), then from those, the one with the highest redundancy value is chosen. In case more than one sensor still satisfies

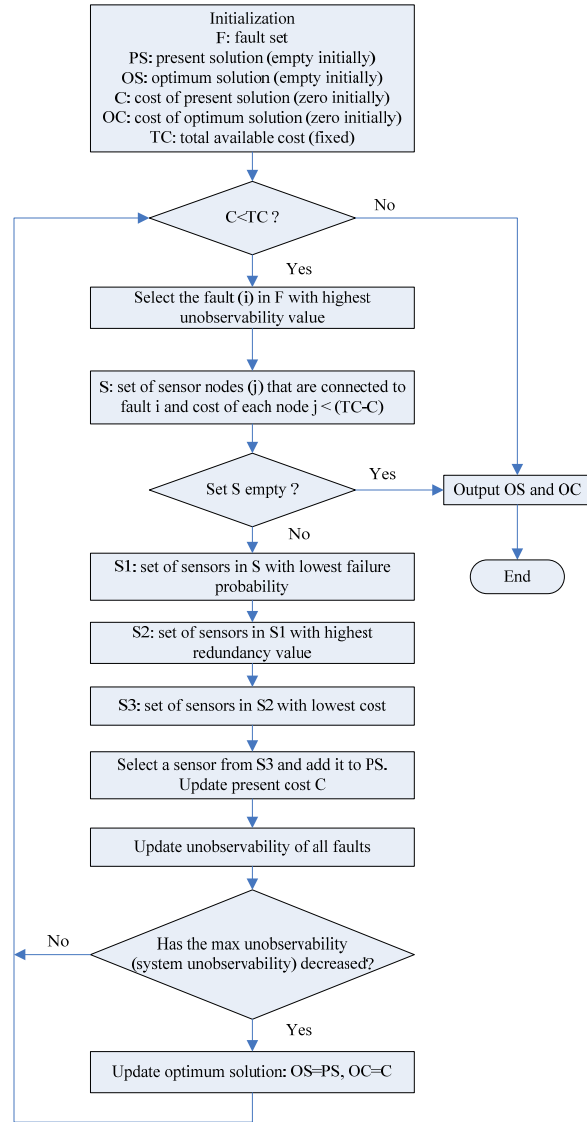


Figure 6.3. Flowchart for greedy search algorithm.

this criterion, the one with the least cost is chosen. If still there is more than one candidate, any sensor among those may be chosen. A crucial point is that no restriction is placed on the number of sensors that can be placed to observe the same node (variable). Hence, it is possible that at some nodes more than one sensor is placed, while some other nodes are not selected as sensor nodes at all. Then perform the updates on PS , C , and $TC-C$. After this sensor selection, recalculate the unobservability values of all faults and the system unobservability (maximum unobservability). If the system unobservability is less than the previous value, then update the optimal solution to be the current solution: optimal solution $OS=PS$, optimal cost $OC=C$. Otherwise, go back to step 2.

Step 4: The procedure is continued until the set S becomes empty. This indicates that no more sensors can be selected to decrease the system unobservability. The set of sensors in PS is the selected sensor location. Steps 2 and 3 ensure minimization of the system unobservability. The procedure of updating the optimal solution only if there is a decrease in the system unobservability ensures that the solution is optimal in the lexicographic sense.

Example:

Let us still use the previous example whose bipartite graph is shown in Figure 6.4 to demonstrate the sensor placement algorithm. The probabilities of occurrence of faults are $f_1 = 0.1, f_2 = 0.1, f_3 = 0.01, f_4 = 0.01, \text{ and } f_5 = 0.001$. The probabilities of failures of sensors available to measure the three measurable nodes are $s_1 = 0.1, s_2 = 0.01, \text{ and } s_3 = 0.001$. For the sake of illustration, we shall assume the same cost of 100 for all of the available sensors. The total available cost TC is equal to 400, which means four available sensors in this specific case. Depending on the scenario considered, different optimum sensor locations may be obtained.

(a) Only observability of faults is considered: For this case, the optimal sensor network is $[S_2(3), S_3]$. Then the unobservability values of faults are as follows:

$$U_1 = 1 \times 10^{-7}, U_2 = 1 \times 10^{-7}, U_3 = 1 \times 10^{-8}, U_4 = 1 \times 10^{-11}, \text{ and } U_5 = 1 \times 10^{-6}$$

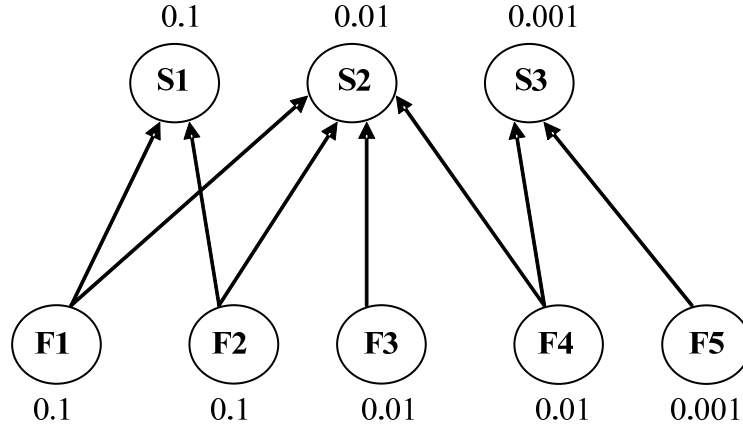


Figure 6.4. Bipartite graph for Figure 6.2 with probabilities.

The system unobservability U is the maximum unobservability. Hence,

$$U = \max(U_1, U_2, U_3, U_4, U_5) = 1 \times 10^{-6}$$

(b) Resolution for the single-fault case is also considered. Virtual faults are constructed exactly like we did in the previous example.

The probabilities of occurrence of these virtual faults are:

$$\begin{aligned} f_{12} &= \min(f_1, f_2) = 0.1 \\ f_{13} &= \min(f_1, f_3) = 0.01 \\ f_{14} &= \min(f_1, f_4) = 0.01 \\ f_{15} &= \min(f_1, f_5) = 0.001 \\ f_{23} &= \min(f_2, f_3) = 0.01 \\ f_{24} &= \min(f_2, f_4) = 0.01 \\ f_{25} &= \min(f_2, f_5) = 0.001 \\ f_{34} &= \min(f_3, f_4) = 0.01 \\ f_{35} &= \min(f_3, f_5) = 0.001 \\ f_{45} &= \min(f_4, f_5) = 0.001 \end{aligned}$$

The virtual faults are then added to the original set of faults. Performing the sensor location algorithm on this extended system gives $[S_1(2), S_2, S_3]$ as the optimum sensor network. With these sensors, the system unobservability is $U = 1 \times 10^{-4}$.

Compared to the previous results based on the fault diagnostic observability and resolution criteria, it is seen that the greedy search-based optimization formulation offers a similar sensor network. The difference lies in the number of sensors that can be put at one location. Because of the additional sensor failure and fault occurrence probability information used in the algorithm, multiple sensors could be placed on any particular nodes in order to minimize the system unobservability, subject to the cost constraints.

6.5 Summary

In many cases, it is found that some faults are still indistinguishable using the sensor set obtained from the above sensor placement scheme. Nonetheless, optimal sensor placement design from a fault diagnosis perspective will provide valuable information to a fault diagnostic system based on principal component analysis, as described later in Chapter 7. The overall method as described in the preceding sections consists of the following steps:

(1) Define all faults of interest in a process (including process fault and sensor fault), based on the operation history records and available process knowledge. Then build DG models of the monitored process, which can be implemented by using empirical relationships or fundamental mathematical model of the process.

(2) Solve the formulated unobservability minimization problem to decide the allocation of sensors. The obtained sensor set would partially guarantee the detection and isolation of all the faults defined in the first step.

(3) Highlight the faults that cannot be isolated by the information provided by the DG models and the sensor network obtained in the steps (1) and (2). Those faults will be left to the PCA-based fault diagnostic system for further detection and isolation.

Chapter 7

Sensor Placement Design and Fault Diagnosis Application for a MSF Desalination System

7.1 Introduction

Principal component analysis (PCA) is a dimensional reduction method, where original data can be represented by a lower dimensional space without significant loss of the variability. From the modeling point of view, PCA transforms correlated variables into uncorrelated ones and determines the linear combinations with large and low variability [Flury, 1989].

Before original data are transformed into a lower dimensional space, they shall be mean-centered because only the variability of the data is of interest. The data are also standardized with unit variance so that equal weights are given to all the variables as far as the variability is concerned. A complete description of the PCA technique is given in this chapter [Jackson, 1991; Jolliffe, 2002]. Also, the sensor placement algorithms developed in Chapter 6, along with the PCA-based fault diagnostic approaches using the optimized sensor set, are demonstrated with application to a MSF desalination system. Both single-fault and dual-fault scenarios are considered in this application.

7.2 Principal Component Analysis for Fault Diagnosis

The basic idea of fault diagnosis using multivariate statistical methods such as PCA is to project the collected data onto a low-dimensional space where the regions of normal operation and abnormal operation can be characterized by fewer state variables.

Because a PCA model represents the variation of normal operation data in a reduced dimensional space, it has better performance of generalization than when the entire measurement space is used. PCA modeling separates the entire measurement space into a model subspace capturing the variation of state variables and a residual subspace containing random variations. The separate characterization of the two subspaces can provide further insights in terms of the

changes in operation conditions. In addition, the linear model extracted by PCA enables us to determine which variables are most affected by a fault and which variables are most responsible for the fault.

The fault detection and isolation (FDI) methodology consists of building a PCA model to characterize the relationships among various measurements. A breakdown in the relationships is indicated as abnormal prediction residuals or lack of fit with the model and may be used for fault detection. Characteristic patterns in the residuals most probably correspond to faults of a particular type and thus may be used for fault isolation. It shall be emphasized that this is true irrespective of the type of a model that is used. More formally, given a model, the detection and isolation of faults may be achieved using the residuals that signify the mismatch between the model predictions and the actual data. It is worth noting that the isolation procedure developed here requires all possible faults are postulated, thus enabling the generation of fault residual sub-spaces.

7.2.1 PCA Algorithm

PCA is a statistical algorithm of dimension reduction by projecting data on to a lower dimensional space such that the major variation of the original data can be preserved. Given a normalized process data matrix X ($m \times n$) composed of m observations with n measured variables. PCA decomposes X into two components, a predicted value $\hat{X}$ and an error value E , which determine two orthogonal subspaces, i.e., the principal component (PC) subspace and the residual component (RC) subspace, respectively.

$$\begin{aligned} X &= \hat{X} + E \\ \hat{X} &= TP^T \\ E &= T_E P_E^T \end{aligned} \tag{7.1}$$

where P is the orthogonal loading matrix and T is the score matrix. The scores T in the PC subspace explain the dominant variation of the measured variables, and the scores T_E in the RC subspace represent the insignificant variation due to model reduction error. The column vectors of principal component loadings P ($n \times l$) are the eigenvectors corresponding to the l largest

eigenvalues of the correlation matrix of the data matrix X and the columns of P_E are the eigenvectors corresponding to the smallest $n-l$ eigenvalues.

The number of PCs retained in a PCA model, l , can be estimated as its corresponding cumulative percentage variance (CPV) greater than a predetermined value (80%, for example) or using algorithms such as the L-curve method. Valle et al. [1999] proposed a new method of variance reconstruction error (VRE) to select the number of PCs and compared it with ten other methods, including Akaike information criterion (AIC), cross validation (based on the PRESS and R ratio), and CPV, etc. Here the CPV is adopted for its simplicity and satisfactory performance. For more complicated applications, other advanced techniques, i.e. VRE, AIC, and cross validation, might be used.

Singular value decomposition (SVD) of the correlation matrix of X can be easily used to obtain the PCs. This technique is briefly described below.

A singular value and corresponding singular vector of a matrix X ($m \times n$) are a scalar δ and a pair of vectors u and v that satisfy:

$$X * v = \delta * u \quad (7.2)$$

By comparison, an eigenvalue λ and the corresponding eigenvector e of an $n \times n$ square matrix $A = X^T X$ is defined by:

$$A * e = \lambda * e \quad (7.3)$$

The corresponding matrices of singular values of X and the eigenvalues of A are written as

$$\Sigma = \begin{bmatrix} \delta_1 & 0 & \dots & 0 \\ 0 & \delta_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \delta_n \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 \end{bmatrix}, \quad D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{bmatrix} \quad (7.4)$$

The corresponding singular vectors form the columns of two orthogonal matrices U and V , and the eigenvectors of the input matrix X form the matrix E :

$$\begin{aligned} U &= [u_1, u_2, \dots, u_m] \\ V &= [v_1, v_2, \dots, v_n] \\ E &= [e_1, e_2, \dots, e_n] \end{aligned} \tag{7.5}$$

The matrices X and A are written as

$$X = U * \Sigma * V^T \tag{7.6}$$

$$A = E * D * E^T \tag{7.7}$$

Since U and V are orthogonal matrices, Equation (7.6) becomes the singular value decomposition, and Equation (7.7) is the eigenvalue decomposition.

The singular value decomposition of an $m \times n$ matrix X results in an $m \times m$ U matrix, an $m \times n$ matrix Σ , and an $n \times n$ matrix V . The column vectors of V are the principal components or eigenvectors of the input matrix X , and V is the same as the matrix E .

Since

$$\begin{aligned} X^T X &= (U \Sigma V^T)^T (U \Sigma V^T) = V \Sigma^T U^T U \Sigma V^T = V \Sigma^T \Sigma V^T \\ A &= X^T X = E D E^T \quad \text{and} \quad E = V \end{aligned}$$

Then

$$D = \Sigma^T \Sigma$$

Therefore, we can always get the same principal components for a matrix using either the SVD technique or through the eigenvalue decomposition of the covariance matrix.

In general, the eigenvalue decomposition is the appropriate tool for analyzing a matrix when it represents a mapping from a vector space onto itself, as it does for an ordinary differential equation. On the other hand, the singular value decomposition is appropriate for analyzing a mapping from one vector space onto another vector space, possibly with a different dimension.

7.2.2 Fault Detection

Fault detection can be performed by monitoring the change of the correlation structure of the measured data. Because the variation of data is separated in the PC subspace and RC subspace, two statistics, namely T^2 and Q , are defined to measure the variation in the two spaces, respectively. If a new observation exceeds the effective region in the PC space defined by the normal operation data, a change in operation regime can be detected. If a significant residual is observed in the RC space, a special event, either due to disturbance changes or due to changes in the relationship between variables, can be detected.

The T^2 statistic measures the variation in the PC subspace, which is defined as:

$$T_i^2 = t_i \lambda^{-1} t_i^T = x_i P \lambda^{-1} P^T x_i^T \leq \delta_T^2 \quad (7.8)$$

where λ is a diagonal matrix of the first l eigenvalues of the correlation matrix of X , which are associated with the PCs retained in the model. $t_i = x_i P$ is referred to as the principal score vector on the i_{th} eigenvector of the PCA model. P is the set of PCs kept in the model and x_i is the i_{th} observation. Here δ_T^2 is the control (or confidence) limit of the T^2 statistic.

The T^2 statistic may be oversensitive when some of the eigenvalues of the correlation matrix are close to zeros. Another limitation of the T^2 statistic arises from the assumption that the raw data follows multivariate normal distribution. The assumption is true only when the normal operation data are collected at one operating condition. However, the normal operation data are usually collected under different operating conditions for an engineering process where many operation modes are possible. Therefore, one shall be cautious when the T^2 statistic is used for process monitoring.

The statistic used in the RC subspace is the Q statistic, or the squared prediction error (SPE), which is defined as:

$$Q_i = e_i e_i^T = x_i (I - PP^T) x_i^T \leq \delta_Q^2 \quad (7.9)$$

where δ_Q^2 is the control limit of the Q index.

The Q statistic quantifies the lack of fit between the sample and the model and denotes the distance of the sample from the nominal operation surface. Under the standard assumptions of a multivariate normal distribution for error matrix E , control limits at confidence level α may be obtained using the χ^2 distribution as [Jackson and Mudholkar, 1979]:

$$Q_\alpha = \Theta_1 \left[\frac{c_\alpha \sqrt{2\Theta_2 h_0^2}}{\Theta_1} + 1 + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} \right]^{1/h_0}, \quad (7.10)$$

where

$$\Theta_i = \text{trace} \left(\frac{EE^T}{I-1} \right)^i \text{ for } i=1,2,3$$

$$h_0 = 1 - \frac{2\Theta_1 \Theta_3}{3\Theta_2^2}$$

And c_α is the standard normal deviate (2.57 for $\alpha = 0.01$). In the above equations, i is the number of samples in the normal operation set and Θ_i is the sum of the i_{th} power of the singular values of $E/(I-1)^{0.5}$. For normal operation, the Q statistic would be very small. Detection of faults is accomplished as violations of the control limit, that is, when Q exceeds Q_α . Because the assumption that the residual vector follows multivariate normal distribution is much more relaxed than multivariate normal distribution of the original measured data, the Q statistic is more appropriate than the T^2 statistic for process monitoring.

7.2.3 Fault Isolation

Building a model to characterize the relationships among the various measurements is an essential part of the FDI methodology. Patterns of the residuals that signify the mismatch between the model and the actual data most likely correspond to faults of a particular type. Specifically, different faults would cause the corresponding residuals to orient toward different directions. These various prediction error directions are referred as the fault directions, and the particular fault may be isolated as the one with maximum projection on the enumerated set of fault directions. The proposed fault isolation scheme is described as follows [Kaistha and Upadhyaya, 2001].

Let $F = [f_1 f_2 f_3 \dots f_R]$, where $f_1 f_2 f_3 \dots f_R$ are column vectors, denote the fault directions for the various fault scenarios that are observed in an existing database. These fault directions can be extracted from the historical data using clustering techniques. The fault direction f_i in the fault matrix F represents the direction in the residual space for the i_{th} fault such that the samples corresponding to the fault have the maximum projection on f_i . In other words, if E_i denotes the residuals for samples corresponding to the i_{th} fault, the optimization problem is

$$J = \max_{f_i} f_i^T E_i^T E_i f_i \quad (7.11)$$

subject to the constraint

$$f_i^T f_i = 1 \quad (7.12)$$

Using the Lagrangian multiplier and differentiating J with respect to f_i and setting the derivative to zero for maximization,

$$2E_i^T E_i f_i - 2\sigma f_i = 0 \quad (7.13)$$

and

$$E_i^T E_i f_i = \sigma f_i \quad (7.14)$$

The fault direction f_i is thus obtained as the first eigenvector of $E_i^T E_i$. The SVD technique may be used to obtain the eigenvector.

Once the fault matrix F is properly defined, fault isolation is accomplished by calculating the projections onto F and classifying the fault as the one with the maximum projection norm. A fault isolation index for the i_{th} fault is defined as

$$FI_i = 1 - Q_i / Q \quad (7.15)$$

where

$$Q_i = e(I - f_i f_i^T)(I - f_i f_i^T)e^T \quad (7.16)$$

In the above equations, Q_i is the distance of the sample from the origin after subtracting the projection of the residuals on the fault direction f_i . It represents the sum of squares of residuals remaining after removing the contribution from the i_{th} fault direction. The fault isolation index quantifies the fraction of Q that is due to f_i . When a particular type of fault scenario, the j_{th} , occurs, the projection of the residuals on f_j would be very high. In other words, FI_i would be the largest and close to 1 for $i = j$, where i varies from 1 to R . This results in the isolation of the fault from the various existing scenarios.

In historical databases, the number of fault scenarios and the correspondence of the fault samples to the particular fault scenarios are not known a priori. Classification of the residuals must then be used to establish the number of fault scenarios and also the correspondence of the samples to the particular scenarios. Existing classification algorithms [James, 1985] can be suitably adapted for this purpose.

7.3 DG Model Development of a MSF System

A MSF desalination process has been described in Chapter 4. Due to the large size of the MSF plant, it must be appropriately divided into sections before the DG models can be built. The structural decomposition is quite suitable for this case thanks to the similarities among stages. A directed graphical model is built for each stage, which only consists of its process variables. Each DG model is a qualitative model corresponding to a normal stage. It can be obtained either automatically by a computer program using the modular approach or by using the set of equations employed in the quantitative model. The first method builds a DG model by joining the common nodes of the corresponding sub-models (e.g. DGs of valves, pumps, etc). These sub-models are contained in a given model library. In this work, the second method is adopted. That is, DG models are obtained from the equations of the quantitative models, which were described in Chapter 4.

Figure 7.1 illustrates the input and output variables in the brine heater and the first flash stage of the MSF plant; and Table 7.1 lists the definitions of these variables. Figure 7.2 shows the DG model corresponding to the brine heater and the first stage.

Figure 7.3 shows a flash stage with its input and output variables. The definitions of the variables are listed in Table 7.2. The DG model of a generic flash stage i is illustrated in Figure 7.4.

For the sake of illustration, we shall employ a simplified MSF system with two flash stages and a brine heater to demonstrate the developed sensor placement and fault diagnostic approaches. The DG model of this system is shown in Figure 7.5. The yellow F nodes represent the root nodes of the system, each of which corresponds to a fault. The 27 measurements that are available are listed in Table 7.3. And Table 7.4 lists eight fault cases that need to be monitored. The fault diagnosis of the simplified MSF system takes into account the sensor faults, process faults, and controller faults. The four sensor faults are all sensor drifts; the heat transfer degradation faults of flash stages #1 and #2 are considered to be process anomalies in the MSF system. Changes in the set points of the top brine temperature (TBT) controller and the brine level controller of stage #2 are identified as the two controller faults for the desalination process.

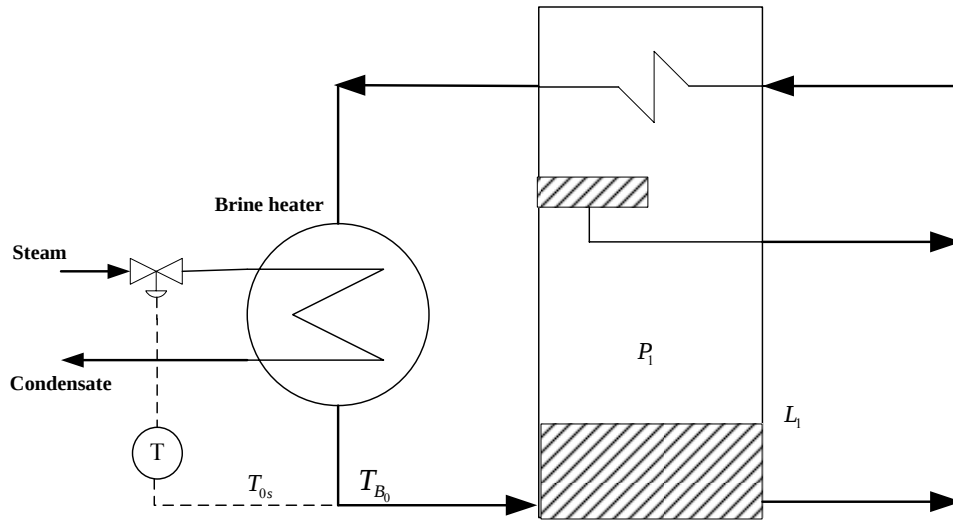


Figure 7.1. Brine heater section of a MSF plant.

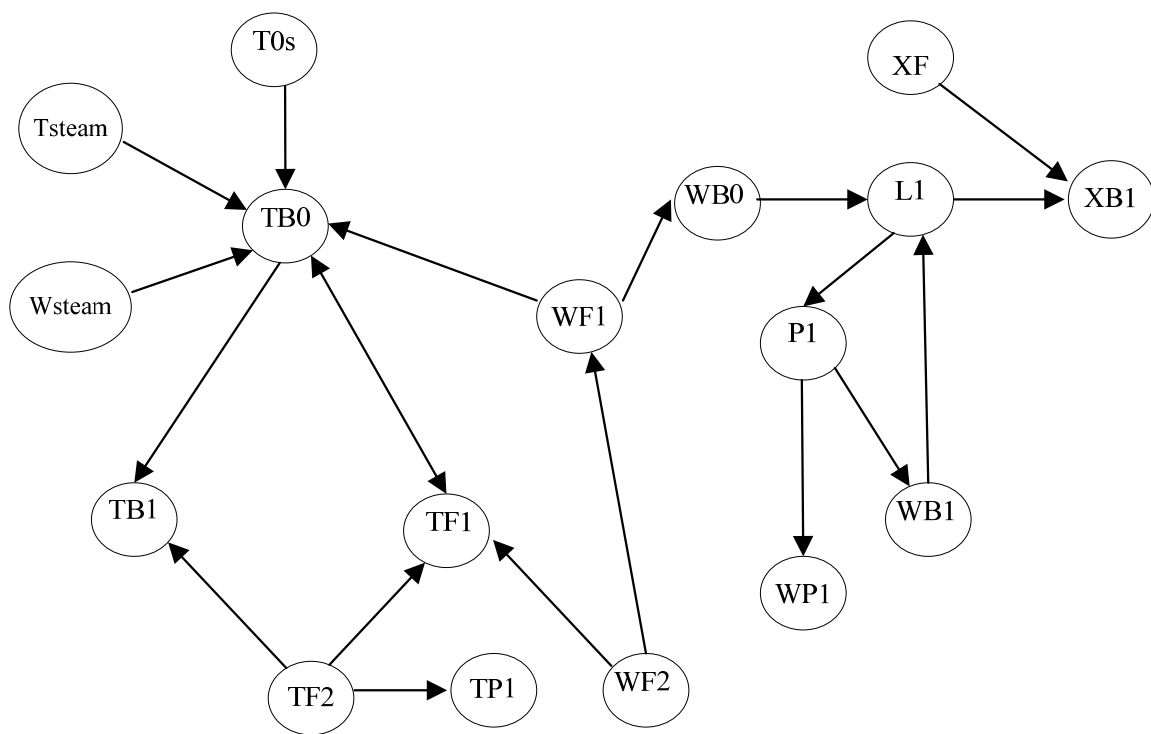


Figure 7.2. Directed graph of brine heater section.

Table 7.1. DG model variables of brine heater

Variable	Definition
T_{STEAM}	Inlet steam temperature
W_{STEAM}	Inlet steam flow rate
T_{B_0}	Top brine temperature
T_{0s}	Top brine temperature set point
T_{B_1}	Flashing brine temperature exiting stage #1
T_{F_1}	Cooling brine temperature exiting stage #1
T_{F_2}	Cooling brine temperature exiting stage #2
T_{R_1}	Distillate product temperature exiting stage #1
W_{F_1}	Cooling brine flow exiting stage #1
W_{F_2}	Cooling brine flow exiting stage #2
W_{B_0}	Cooling brine flow exiting the brine heater
W_{B_1}	Flashing brine flow exiting stage #1
W_{R_1}	Distillate product flow exiting stage #1
X_F	Inlet cooling brine salinity
X_{B_1}	Flashing brine salinity exiting stage #1
L_1	Brine level in stage #1
P_1	Stage #1 pressure

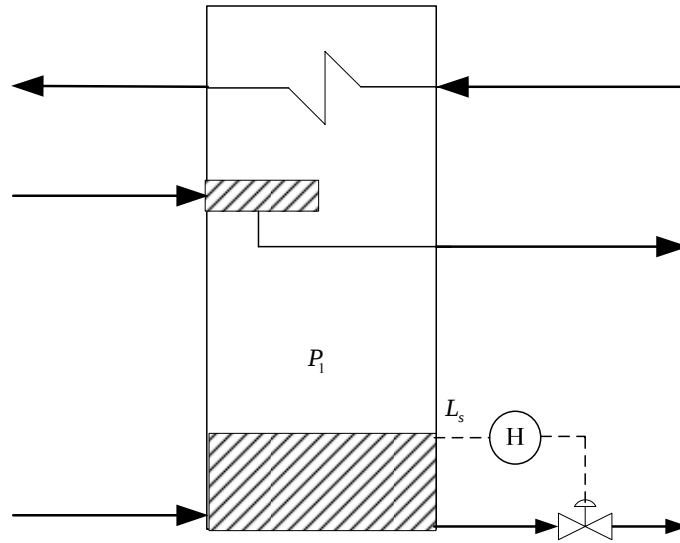


Figure 7.3. A generic flash stage of a MSF plant.

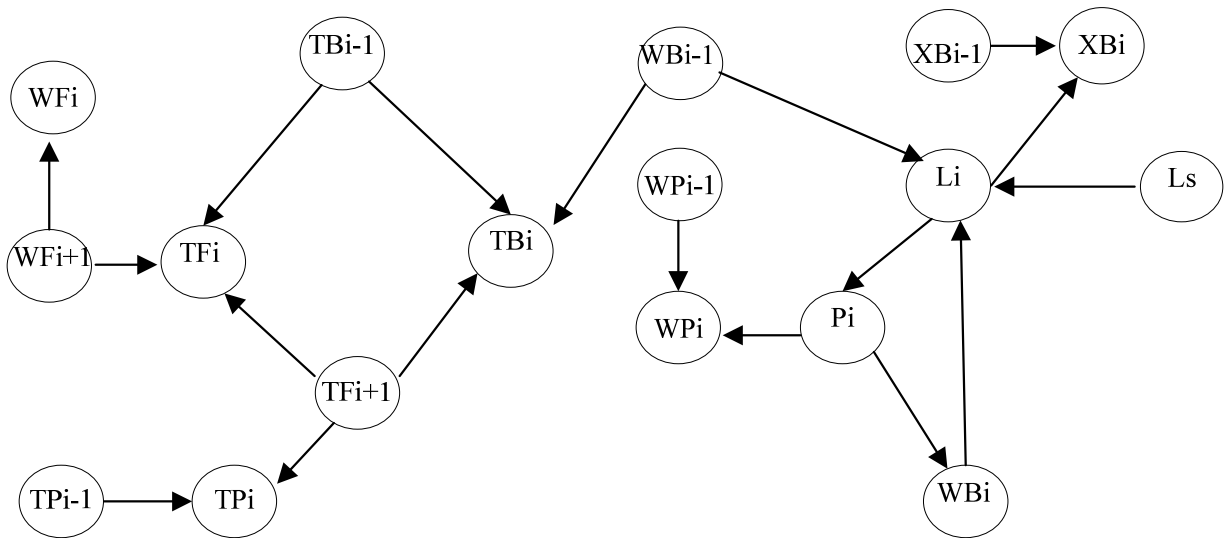


Figure 7.4. Directed graph of a flash stage i .

Table 7.2. DG model variables of flash stage i

Variable	Definition
T_{F_i}	Cooling brine temperature exiting stage i
$T_{F_{i+1}}$	Cooling brine temperature entering stage i
T_{P_i}	Distillate product temperature exiting stage i
$T_{P_{i-1}}$	Distillate product temperature entering stage i
T_{B_i}	Flashing brine temperature exiting stage i
$T_{B_{i-1}}$	Flashing brine temperature entering stage i
W_{F_i}	Cooling brine flow exiting stage i
$W_{F_{i+1}}$	Cooling brine flow entering stage i
W_{P_i}	Distillate product flow exiting stage i
$W_{P_{i-1}}$	Distillate product flow entering stage i
W_{B_i}	Flashing brine flow exiting stage i
$W_{B_{i-1}}$	Flashing brine flow entering stage i
X_{B_i}	Flashing brine salinity exiting stage i
$X_{B_{i-1}}$	Flashing brine salinity entering stage i
L_i	Brine level in stage i
L_s	Brine level set point
P_i	Stage i pressure

Table 7.3. Two-stage MSF system available variables

Index	Description
S1	Inlet cooling brine temperature
S2	Inlet cooling brine flow
S3	Inlet cooling brine salinity
S4	Inlet steam temperature
S5	Inlet steam flow rate
S6	Distillate product temperature exiting stage #1
S7	Distillate product temperature exiting stage #2
S8	Flashing brine temperature exiting stage #1
S9	Flashing brine temperature exiting stage #2
S10	Cooling brine temperature exiting stage #1
S11	Cooling brine temperature exiting stage #2
S12	Flashing brine flow exiting stage #1
S13	Flashing brine flow exiting stage #2
S14	Flashing brine salinity exiting stage #1
S15	Flashing brine salinity exiting stage #2
S16	Top brine temperature
S17	Top brine temperature controller set point
S18	Distillate product flow exiting stage #1
S19	Distillate product flow exiting stage #2
S20	Stage #1 pressure
S21	Stage #2 pressure
S22	Brine level in stage #1
S23	Brine level in stage #2
S24	Stage #2 brine level controller set point
S25	Cooling brine flow exiting the brine heater
S26	Cooling brine flow exiting stage #1
S27	Cooling brine flow exiting stage #2

Table 7.4. Faults considered for two-stage MSF system

Fault Nodes	System Faults
F1	Inlet cooling brine temperature sensor drift
F2	Inlet cooling brine flow sensor drift
F3	Inlet cooling brine salinity sensor drift
F4	Flashing stage #1 heat transfer degradation
F5	Flashing stage #2 heat transfer degradation
F6	Top brine temperature sensor drift
F7	Top brine temperature controller fault
F8	Stage #2 brine level controller fault

7.4 Sensor Placement Design

Both the greedy search heuristic and LINGO 8.0 are used to find the minimum set of sensors required to observe all the eight faults listed in Table 7.4 for the two-stage MSF system. This is a set-covering problem that has been discussed in Chapter 6. This optimization problem can be formulated as follows:

$$\text{minimize } \left[\sum_{i=1}^{27} x_i \right] \quad (7.17)$$

subject to

$$\sum_{i=1}^{27} D_{ij} x_i \geq 1, \quad j = 1, \dots, 8 \quad (7.18)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, 27 \quad (7.19)$$

Equation (7.17) is the objective function of the set-covering problem, where x_i is the decision variable. Equation (7.18) is a constraint to ensure that each column (fault) is covered by at least one row (sensor location) where D_{ij} is the $(i, j)_{th}$ entry of the fault-sensor maximum connectivity matrix whose elements comprise of either '1' or '0'. A row i ($i = 1, \dots, m$) covers a column j ($j = 1, \dots, n$) if $D_{ij} = 1$. Finally, Equation (7.19) allows the decision variables x_i to only take binary numbers.

Both methods give nodes $[S_9, S_{15}, S_{16}, S_{23}]$ as the sensor set. Although all the faults can be detected, not every fault can be distinguished from one another.

To obtain the set of sensors that would give maximum resolution under single-fault assumption, additional virtual faults have to be created as discussed in Chapter 6. Sets A_i are associated with the original faults. The virtual faults $B_{ij} = A_i \cup A_j - A_i \cap A_j$ are also constructed. This involves generation of $C_8^2 = 28$ virtual faults, so that the system now has 36 faults. Now each B_{ij} is represented as a fault node, and a bipartite graph is constructed between these nodes and the sensor nodes. The new bipartite graph is added to the original bipartite graph, and the new optimization problem can be modified as follows:

$$\text{minimize } \left[\sum_{i=1}^{27} x_i \right] \quad (7.20)$$

subject to

$$\sum_{i=1}^{27} D_{ij} x_i \geq 1, \quad j = 1, \dots, 36 \quad (7.21)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, 27 \quad (7.22)$$

The greedy search based heuristic presented for fault diagnostic observability criterion is applied to the new problem. This gives $[S_1, S_2, S_3, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{15}, S_{16}, S_{17}, S_{23}, S_{24}]$ as the minimum sensor set for full isolation of the selected faults.

For the dual-fault case, solving the sensor location problem is done by assuming that two simultaneous faults can occur along with the possibility of the occurrence of a single fault. Under this assumption, the sets $A_{ij} = A_i \cup A_j$ are formed for all faults. Each new virtual fault is treated as a root node along with the original faults. The above greedy algorithm gives $[S_1, S_2, S_3, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{15}, S_{16}, S_{17}, S_{23}, S_{24}]$ as the set of sensor nodes. An interesting point to note is that the minimum number of sensors for solving single-fault and dual-fault resolution problems turns out to be the same. This would mean that dual-fault resolution cannot be improved with adding more sensors. This is a result that is not obvious from the DG of the process.

7.5 PCA Model Development of a MSF System

In the previous section, the design problem of sensor network with maximum fault resolution using fault observability criterion has been presented. The greedy search algorithm gives $[S_1, S_2, S_3, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{15}, S_{16}, S_{17}, S_{23}, S_{24}]$ as the minimum sensor set for fully isolation of the identified faults. The advantage of using this optimized sensor set is that more information about the system is utilized, and some basic properties such as the fault detectability and identifiability are already partially guaranteed before PCA has been employed to monitor system behavior.

7.5.1 Database Generation

A normal operation database was generated using the developed two-stage MSF SIMULINK model. The TBT controller set point and the brine level controller set point in the last flashing stage (Stage #2) were systematically changed one at a time. About 1,728 cases were simulated and the data generated were stored in a database. The list of the measured variables used to develop the PCA model is given in Table 7.5.

7.5.2 PCA Modeling

A PCA model is built using the data for the nominal operation case. The nominal operation data matrix is preprocessed by auto-scaling the columns in the data matrix to zero mean and unit variance. This puts all the measurements with their different units on a common unit variance scale. The percentage information explained by each PC is shown in Figure 7.6. Five principal components are retained, and they explain ~99.9% of the total variation in the data, indicating the high degree of redundancy in the measurements. The T^2 and Q statistics with 95% confidence level for the samples in the fault-free database are plotted in Figure 7.7 and Figure 7.8, respectively. If the corresponding statistics exceeds the limit, the confidence to state that the fault free model cannot explain the data is at a level greater than 95%. The two figures illustrate that all the fault free data are well below the limit lines. The probability of false alarms due to process disturbance is low.

Table 7.5. MSF variables used to develop PCA models

Variable	Description
1	Inlet cooling brine temperature (S1)
2	Inlet cooling brine flow (S2)
3	Inlet cooling brine salinity (S3)
4	Flashing brine temperature exiting stage #1 (S8)
5	Flashing brine temperature exiting stage #2 (S9)
6	Cooling brine temperature exiting stage #1 (S10)
7	Cooling brine temperature exiting stage #2 (S11)
8	Flashing brine flow exiting stage #1 (S12)
9	Flashing brine salinity exiting stage #2 (S15)
10	Top brine temperature (S16)
11	Top brine temperature controller set point (S17)
12	Brine level in the stage #2 (S23)
13	Stage #2 brine level controller set point (S24)

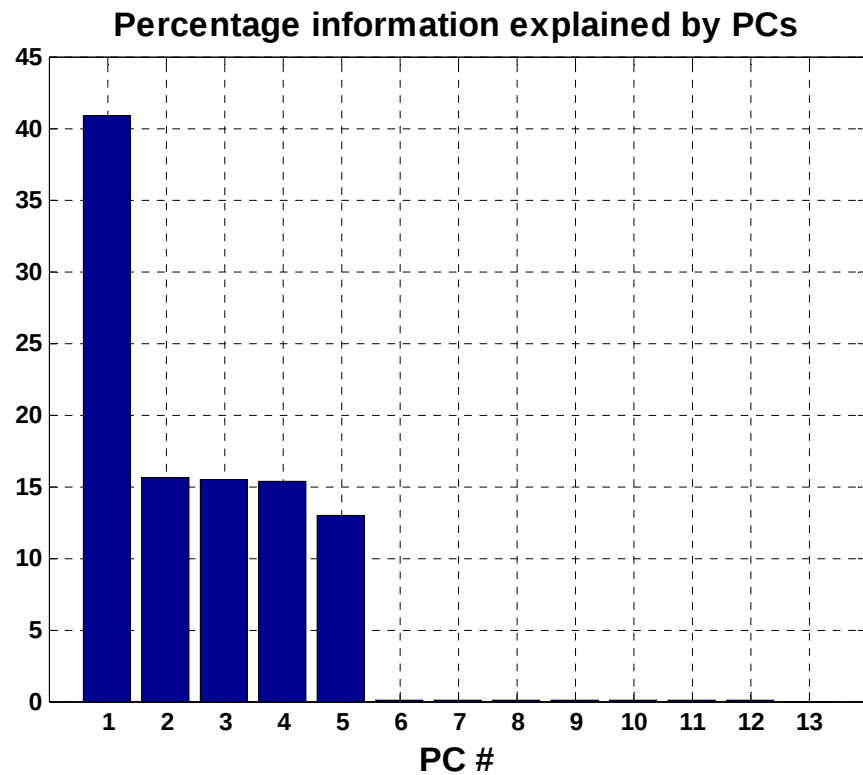


Figure 7.6. Percentage of variance explained by principal components.

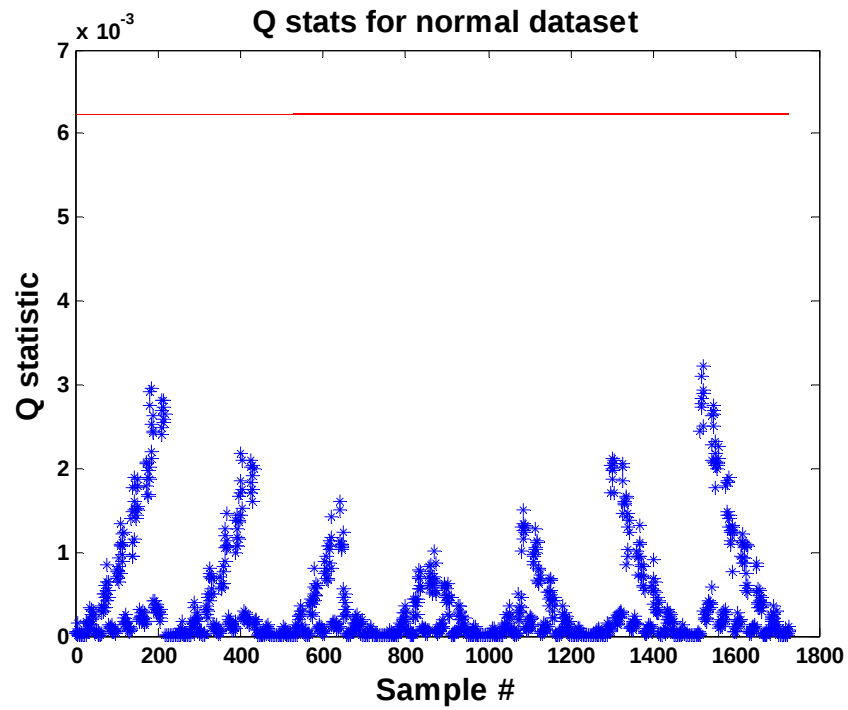


Figure 7.7. Q statistic for MSF PCA model.

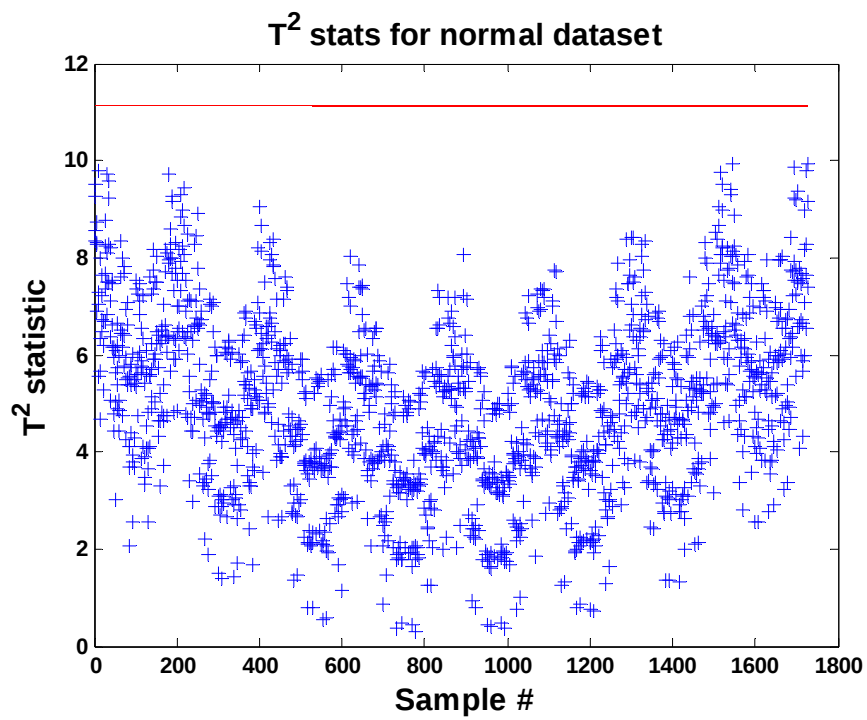


Figure 7.8. T^2 statistic for MSF PCA model.

7.6 Fault Detection Results

Under normal conditions, small residuals would be generated and limited to a certain range. In the test cases when one or several components in the system are under degradation, the so-called causal relations among these variables will be violated. As a result, the mapping of residuals from residual generators or system models will increase in a specific direction. As an example, Figure 7.9 (a) shows the residual patterns from the PCA models when there is a drift of 0.2% to 1% nominal value in the inlet cooling brine temperature sensor (Fault #1). The residuals are the differences between the measurement values and their PCA model predictions. Figure 7.9 (b) shows a significant increase in the Q statistic of the PCA model exceeding the 95% confidence level. The T^2 statistic for the faulty conditions stays within the range, as in Figure 7.9 (c). This would be categorized as the fault scenario where the Q statistic is outside the limits and the T^2 statistic is within the limits [Humberstone, 2010]. These illustrate the capability of data-driven models in detecting system anomalies. It should be kept in mind that both T^2 and Q statistics must be used for fault detection. Either statistic being violated will signify that a fault has happened. Violation of the T^2 statistic represents that the system operates at an abnormal state beyond the model space, while departure of the Q statistic represents that some of the constraint equations defined in the residual space are violated and the system is abnormal.

It is clear that the residuals reflect not only whether there is an abnormal component, but also the severity of the fault, which is very important in helping the operator or the automatic controller to select the correct strategy in order to avoid severe negative effect caused by faulty devices. The residual patterns for the other seven faults are shown in the following figures (Figure 7.10-7.16).

It should be noted that PCA can only deal with steady-state condition or a slow dynamic process. The algorithm to perform PCA based fault detection is only applicable to steady-state condition. The confidence level will affect the false alarm. In a real application, the confidence level needs to be adjusted according to the operation requirements.

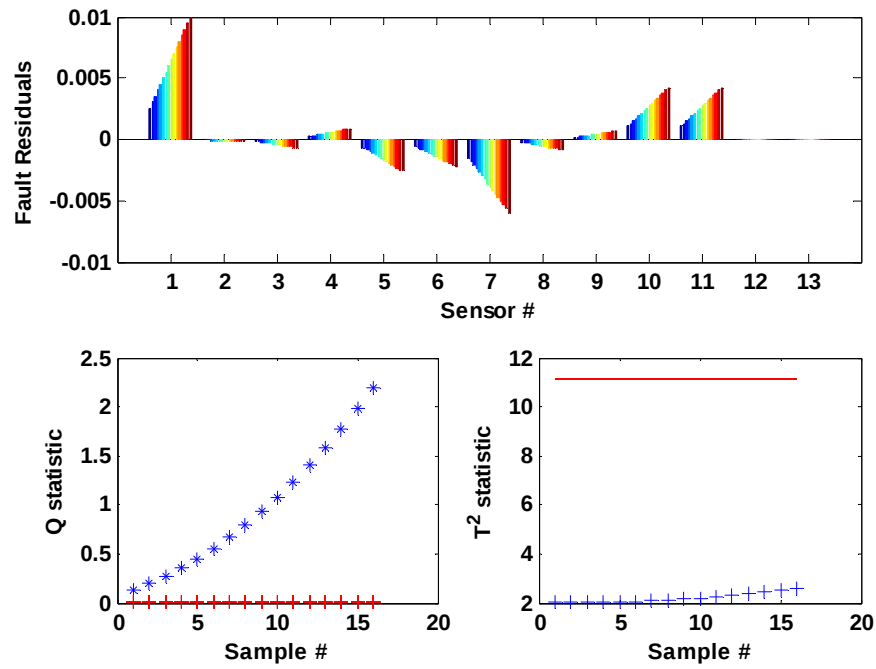


Figure 7.9. Residual pattern for inlet cooling brine temperature sensor drift (Fault #1).

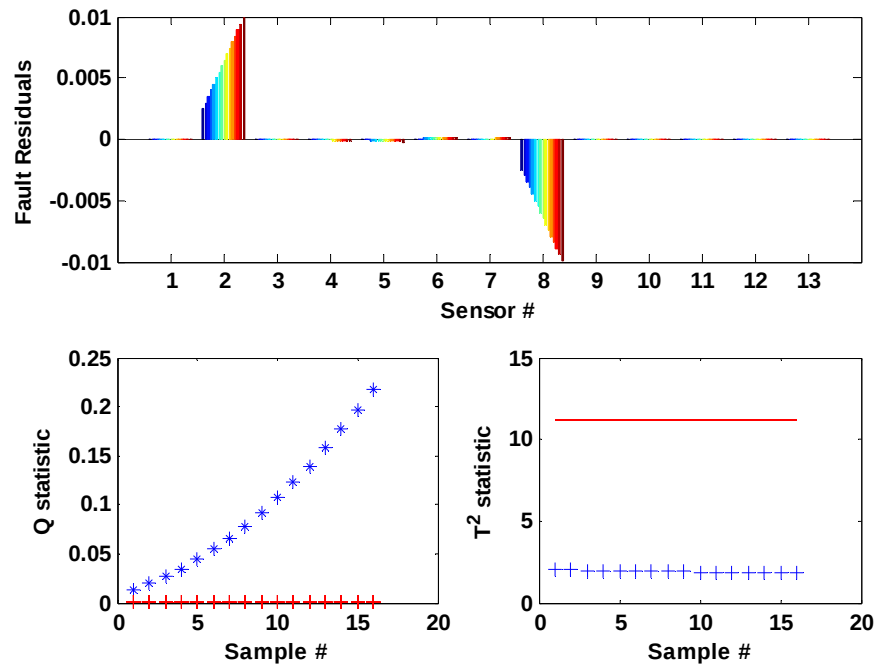


Figure 7.10. Residual pattern for inlet cooling brine flow sensor drift (Fault #2).

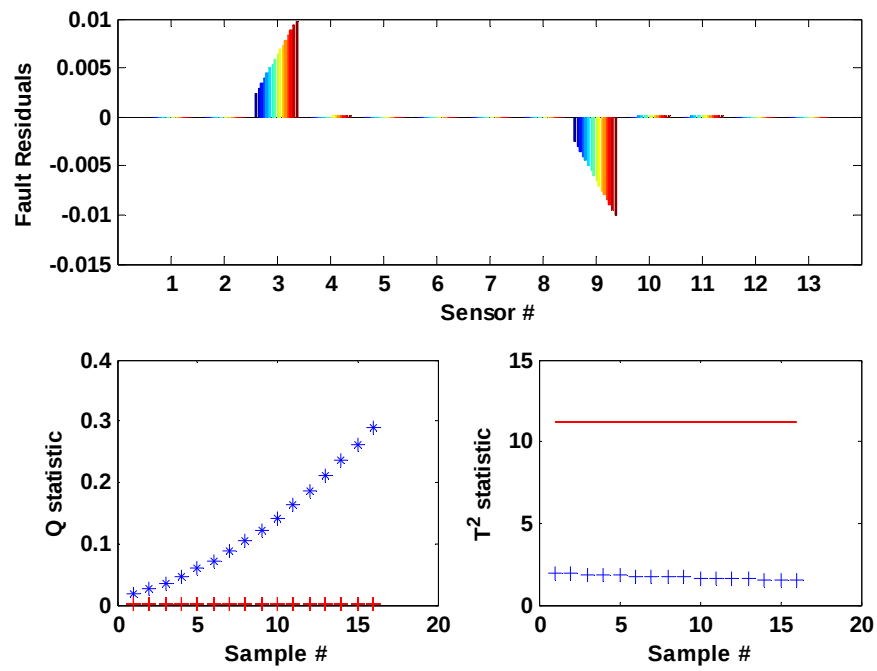


Figure 7.11. Residual pattern for inlet cooling brine salinity sensor drift (Fault #3).

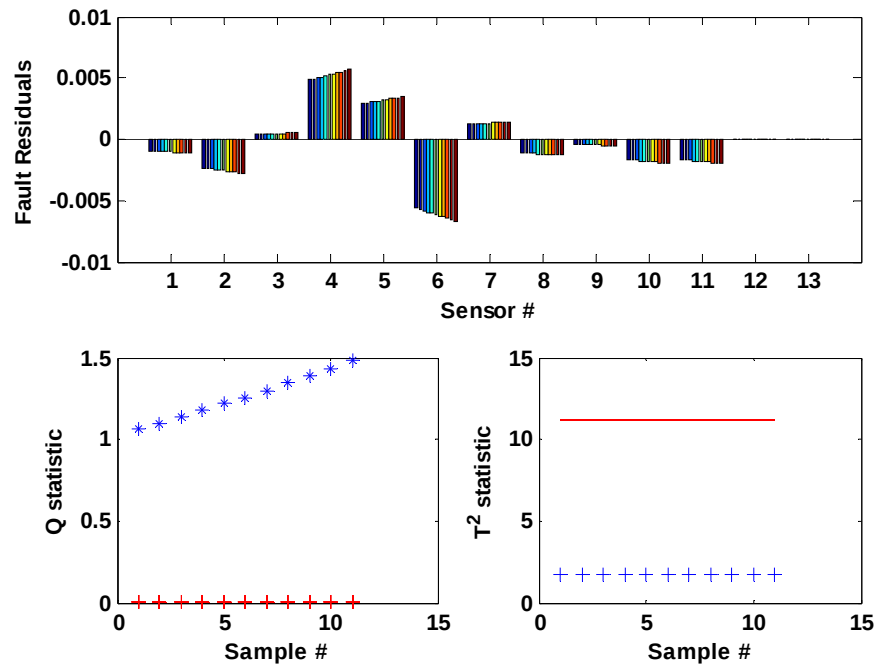


Figure 7.12. Residual pattern for flashing stage #1 heat transfer degradation (Fault #4).

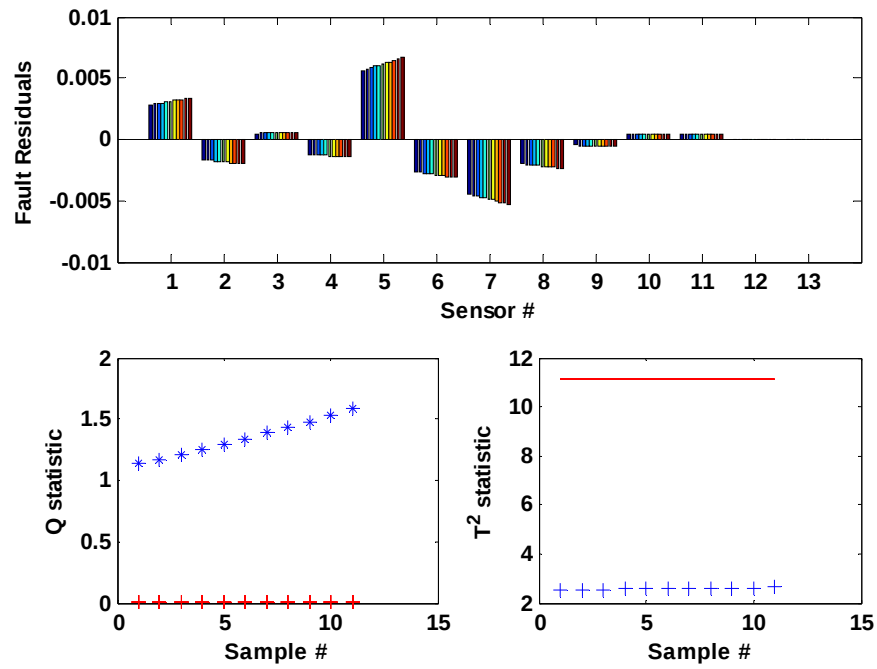


Figure 7.13. Residual pattern for flashing stage #2 heat transfer degradation (Fault #5).

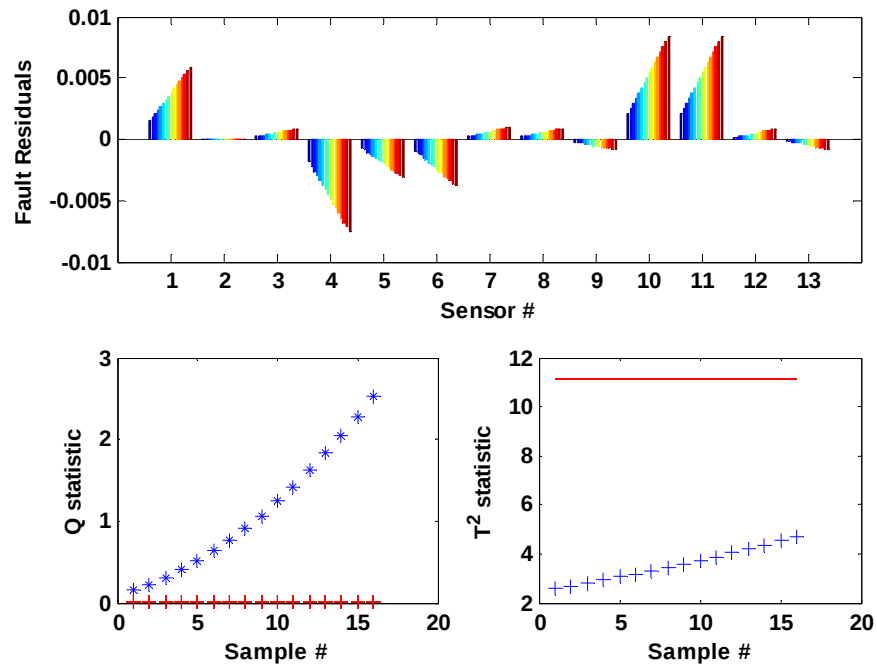


Figure 7.14. Residual pattern for top brine temperature sensor drift (Fault #6).

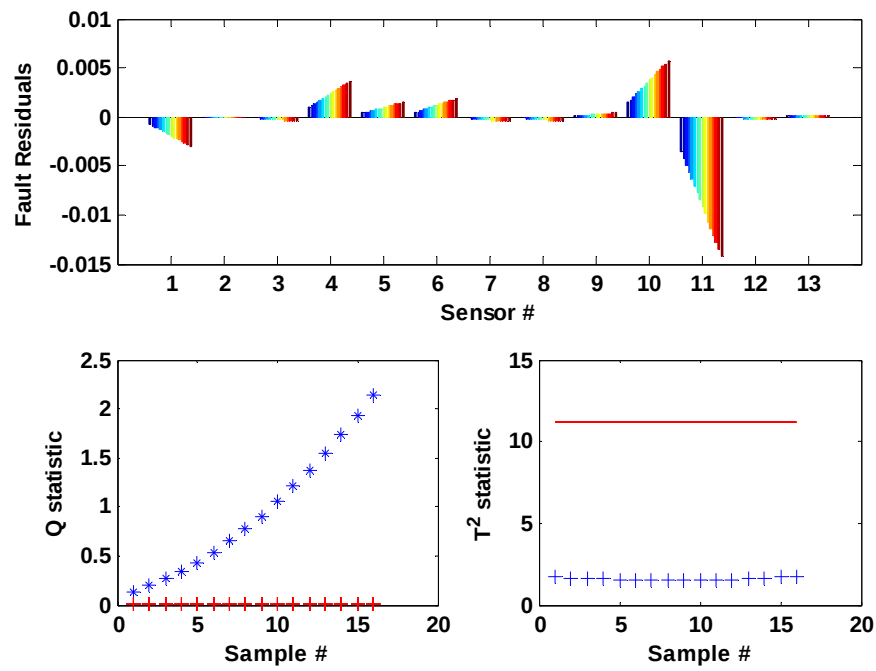


Figure 7.15. Residual pattern for top brine temperature controller fault (Fault #7).

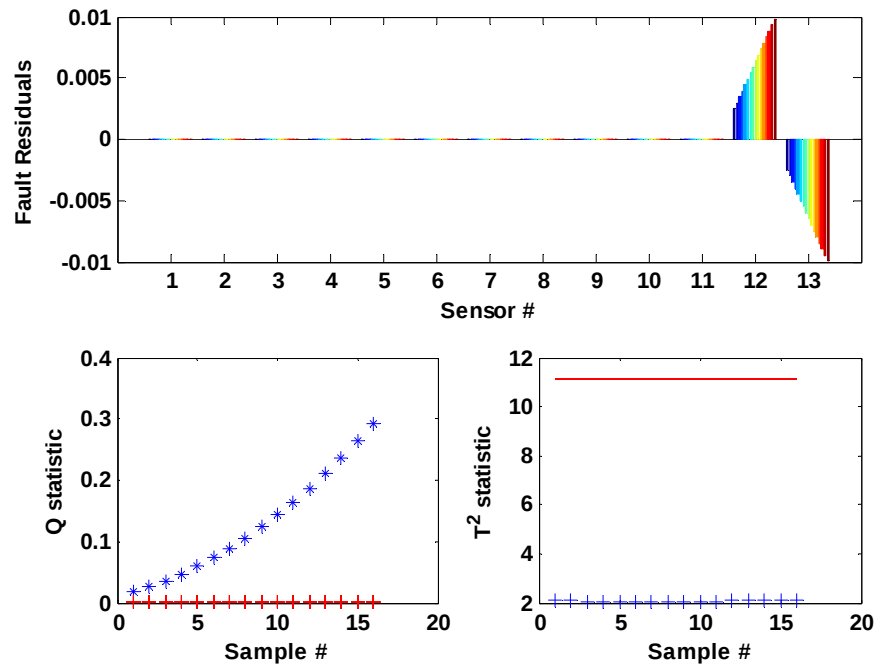


Figure 7.16. Residual pattern for Stage #2 brine level controller fault (Fault #8).

7.7 Fault Isolation Results

Under normal conditions, small residuals would be generated and limited to a certain range. In the test cases when one or several components in the system are under degradation, the causal relations among the variables will be violated. As a result, the mapping of residuals from residual generators or system models will increase in a specific direction.

7.7.1 Single-Fault Cases

The PCA fault direction analysis is performed through processing the fault residuals. The eight fault directions which correspond to the eight fault scenarios considered for the MSF system are given in Table 7.6.

The following plots (Figures 7.17 - 7.24) illustrate the fault isolation indices for the eight fault cases. For instance, Figure 7.17 shows the fault isolation index for the inlet cooling brine temperature sensor drift (Fault #1) scenario in the MSF system. Note that the fault index is close to unity for the particular fault that occurs and is smaller for the others.

Table 7.6. PCA fault direction analysis for MSF system under single fault assumption

Fault direction #	System faults	% of residual variances explained by the 1st PC
1	Inlet cooling brine temperature sensor drift	92.56%
2	Inlet cooling brine flow sensor drift	88.29%
3	Inlet cooling brine salinity sensor drift	94.45%
4	Flashing stage #1 heat transfer degradation	93.68%
5	Flashing stage #2 heat transfer degradation	91.15%
6	Top brine temperature sensor drift	89.47%
7	Top brine temperature controller fault	86.44%
8	Stage #2 brine level controller fault	87.96%

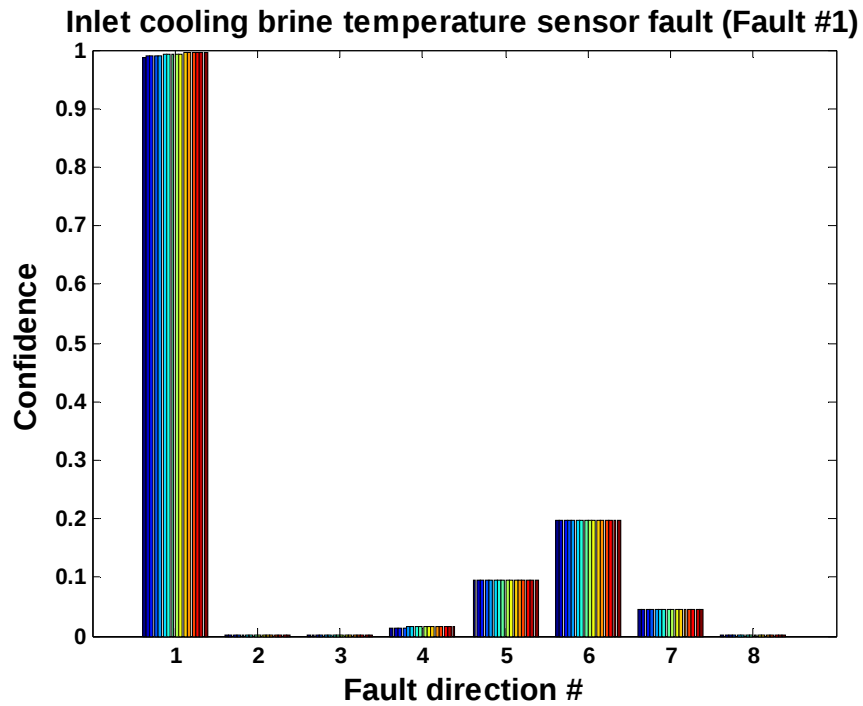


Figure 7.17. Fault isolation index for inlet cooling brine temperature sensor drift.

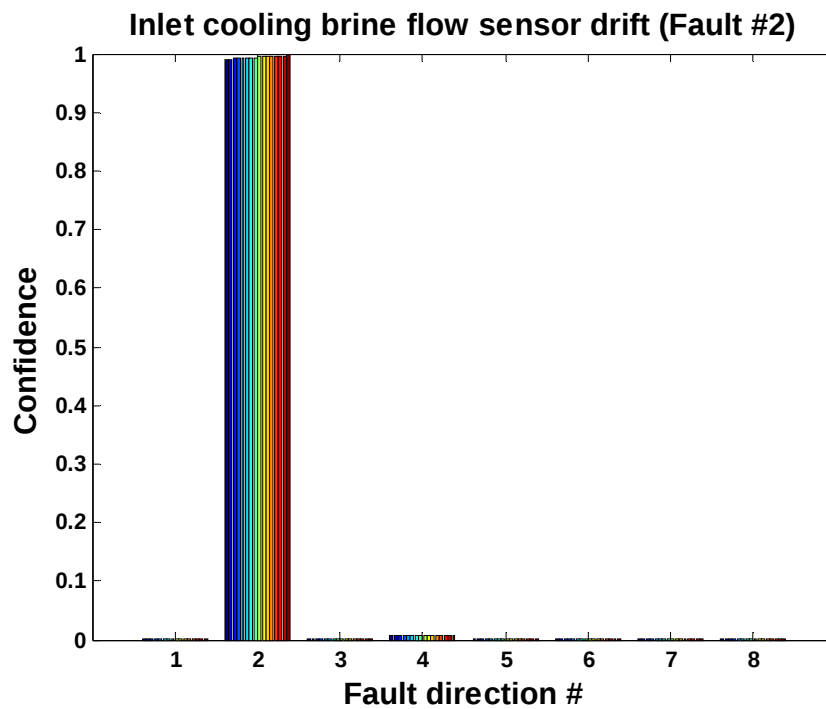


Figure 7.18. Fault isolation index for inlet cooling brine flow sensor drift.

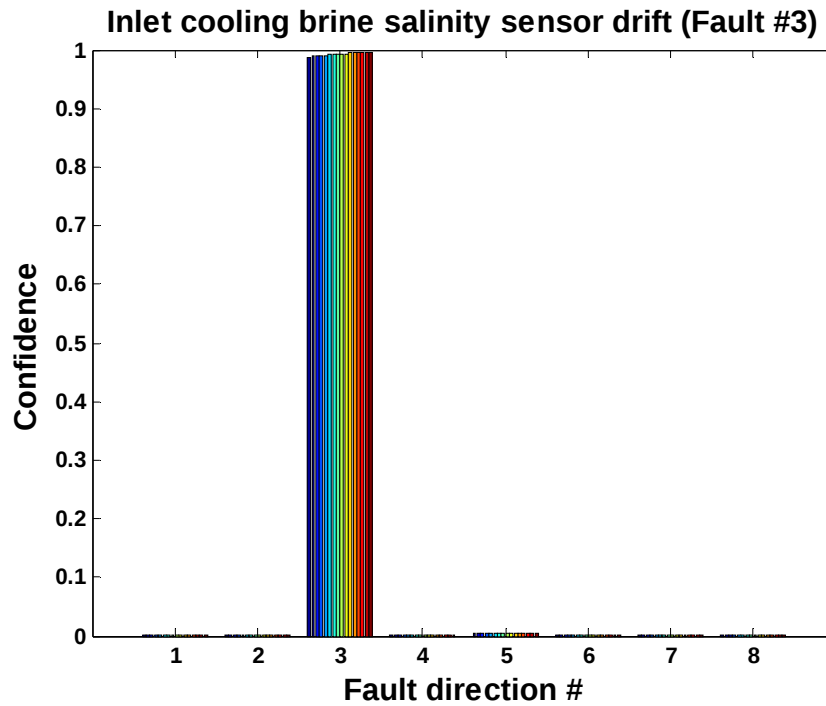


Figure 7.19. Fault isolation index for inlet cooling brine salinity sensor drift.

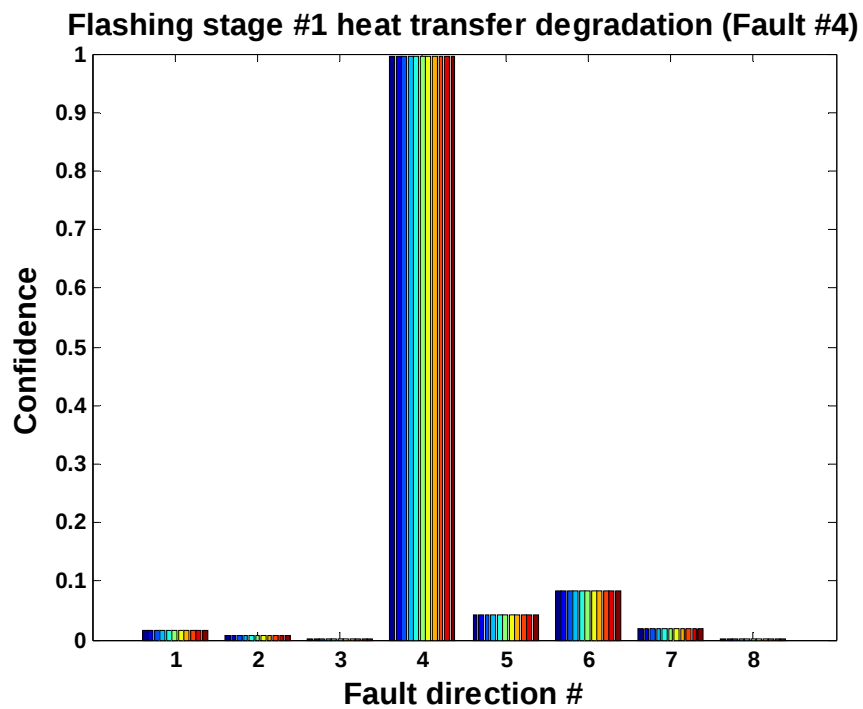


Figure 7.20. Fault isolation index for flashing stage #1 heat transfer degradation.

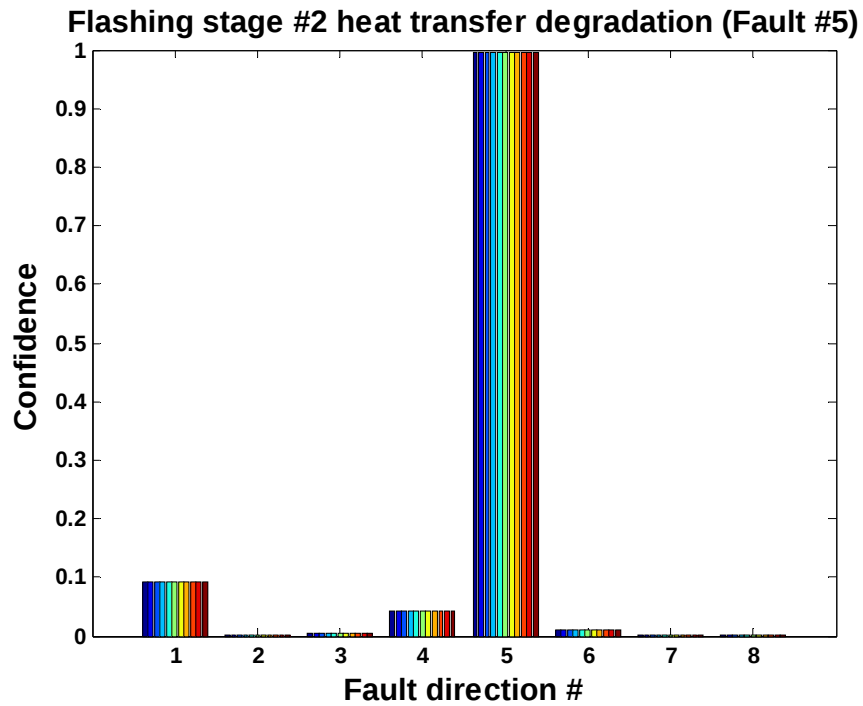


Figure 7.21. Fault isolation index for flashing stage #2 heat transfer degradation.

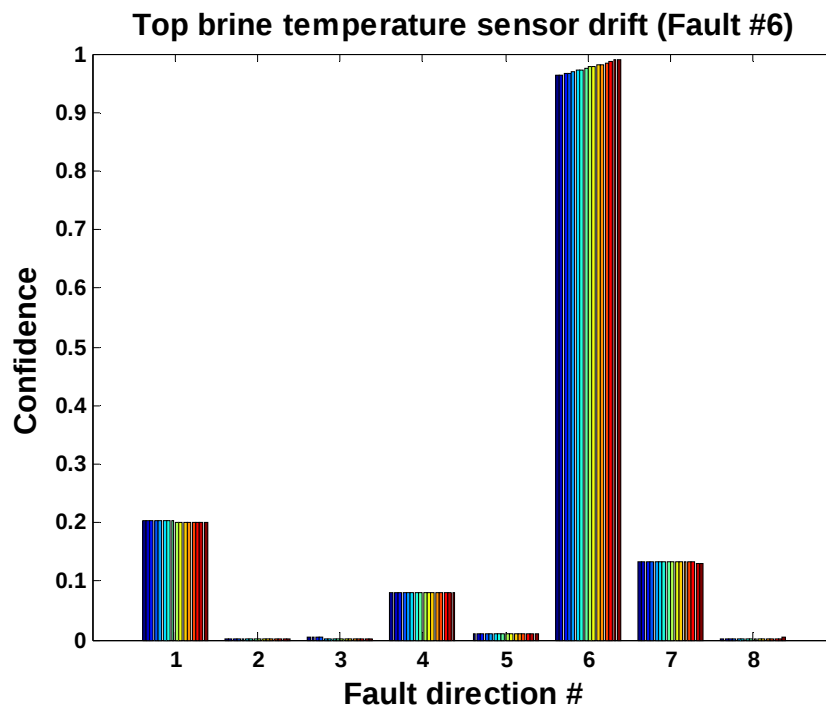


Figure 7.22. Fault isolation index for top brine temperature sensor drift.

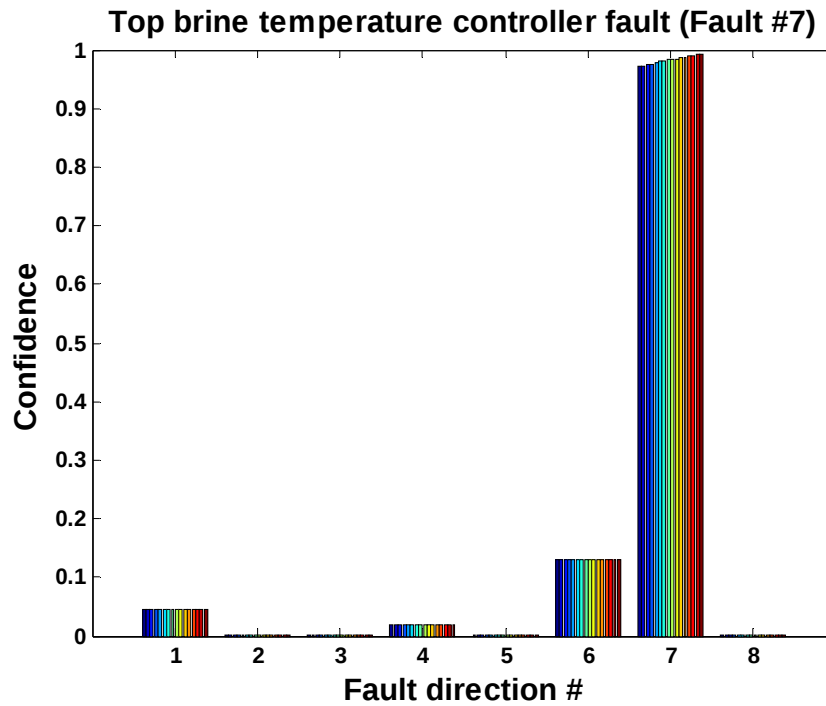


Figure 7.23. Fault isolation index for top brine temperature controller fault.

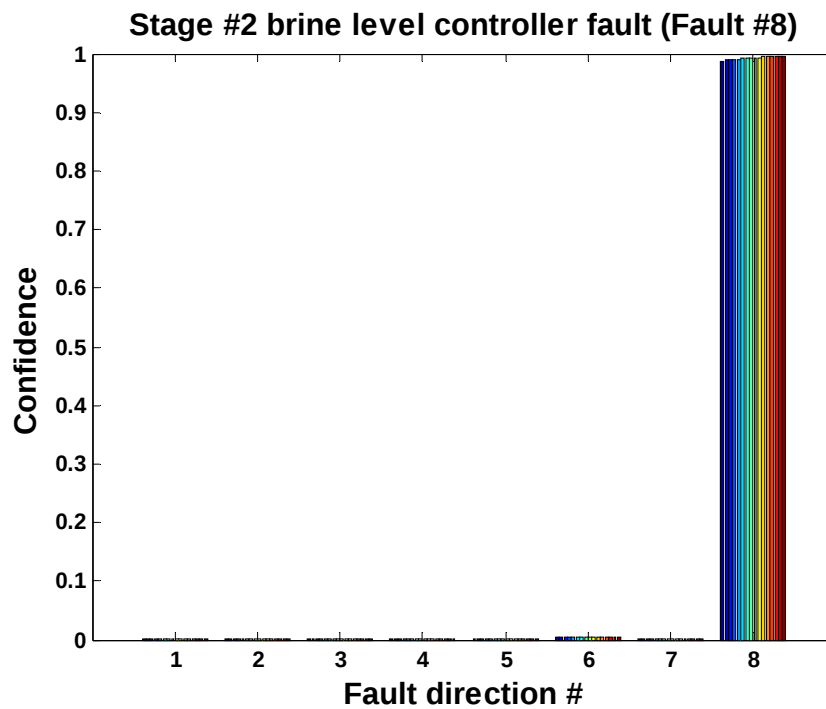


Figure 7.24. Fault isolation index for Stage #2 brine level controller fault.

As can be seen from the above figures, all fault cases are diagnosed correctly using the developed fault isolation approach. Thus, PCA provides a convenient method in dealing with directional residuals. The important assumption that should be emphasized is that the faulty residuals are projected onto the first eigenvector of the residual covariance matrix without losing the direction. In other words, the first PC provides most of the information about the faulty device. However, one shall be cautious about this assumption since it is not always satisfied, especially during a transient of the system operation. Nonetheless, the PCA algorithm demonstrates its effectiveness for fault detection and isolation in complex systems with the optimized sensor allocation, and therefore deserves further studies.

7.7.2 Dual-Fault Cases

The PCA based directional approach can be extended to detect and isolate multi-dimensional faults such as simultaneous multiple faults. The application to the two-stage MSF system is presented in this section. It is assumed that the possible simultaneous faults are limited to dual faults for all the considered faults in the system. The previous work that is publicly available on simultaneous multiple fault detection is rather limited [Watanabe, 1994]. The research results presented in this section are, therefore, original contributions to the area of fault diagnosis.

The same sensor set is applied to detect and isolate the dual faults. As in the single fault cases, the following eight fault directions which correspond to the eight fault scenarios are defined in the designed FDI scheme:

Single fault direction 1: Inlet cooling brine temperature sensor drifting fault

Single fault direction 2: Inlet cooling brine flow sensor drifting fault

Single fault direction 3: Inlet cooling brine salinity sensor drifting fault

Single fault direction 4: Flashing stage #1 heat transfer degradation

Single fault direction 5: Flashing stage #2 heat transfer degradation

Single fault direction 6: Top brine temperature sensor drifting fault

Single fault direction 7: Top brine temperature controller fault

Single fault direction 8: Stage #2 brine level controller fault

The additional dual-fault directions are considered as well:

Dual fault direction 1: Inlet cooling brine temperature sensor drifting fault & Inlet cooling brine flow sensor drifting fault

Dual fault direction 2: Inlet cooling brine temperature sensor drifting fault & Inlet cooling brine salinity sensor drifting fault

Dual fault direction 3: Inlet cooling brine temperature sensor drifting fault & Flashing stage #1 heat transfer degradation

Dual fault direction 4: Inlet cooling brine temperature sensor drifting fault & Flashing stage #2 heat transfer degradation

Dual fault direction 5: Inlet cooling brine temperature sensor drifting fault & Top brine temperature sensor drifting fault

Dual fault direction 6: Inlet cooling brine temperature sensor drifting fault & Top brine temperature controller fault

Dual fault direction 7: Inlet cooling brine temperature sensor drifting fault & Stage #2 brine level controller fault

Dual fault direction 8: Inlet cooling brine flow sensor drifting fault & Inlet cooling brine salinity sensor drifting fault

Dual fault direction 9: Inlet cooling brine flow sensor drifting fault & Flashing stage #1 heat transfer degradation

Dual fault direction 10: Inlet cooling brine flow sensor drifting fault & Flashing stage #2 heat transfer degradation

Dual fault direction 11: Inlet cooling brine flow sensor drifting fault & Top brine temperature sensor drifting fault

Dual fault direction 12: Inlet cooling brine flow sensor drifting fault & Top brine temperature controller fault

Dual fault direction 13: Inlet cooling brine flow sensor drifting fault & Stage #2 brine level controller fault

Dual fault direction 14: Inlet cooling brine salinity sensor drifting fault & Flashing stage #1 heat transfer degradation

Dual fault direction 15: Inlet cooling brine salinity sensor drifting fault & Flashing stage #2 heat transfer degradation

Dual fault direction 16: Inlet cooling brine salinity sensor drifting fault & Top brine temperature sensor drifting fault

Dual fault direction 17: Inlet cooling brine salinity sensor drifting fault & Top brine temperature controller fault

Dual fault direction 18: Inlet cooling brine salinity sensor drifting fault & Stage #2 brine level controller fault

Dual fault direction 19: Flashing stage #1 heat transfer degradation & Flashing stage #2 heat transfer degradation

Dual fault direction 20: Flashing stage #1 heat transfer degradation & Top brine temperature sensor drifting fault

Dual fault direction 21: Flashing stage #1 heat transfer degradation & Top brine temperature controller fault

Dual fault direction 22: Flashing stage #1 heat transfer degradation & Stage #2 brine level controller fault

Dual fault direction 23: Flashing stage #2 heat transfer degradation & Top brine temperature sensor drifting fault

Dual fault direction 24: Flashing stage #2 heat transfer degradation & Top brine temperature controller fault

Dual fault direction 25: Flashing stage #2 heat transfer degradation & Stage #2 brine level controller fault

Dual fault direction 26: Top brine temperature sensor drifting fault & Top brine temperature controller fault

Dual fault direction 27: Top brine temperature sensor drifting fault & Stage #2 brine level controller fault

Dual fault direction 28: Top brine temperature controller fault & Stage #2 brine level controller fault

Both single-fault directions and dual-fault directions are used as fault signature for dual fault isolation. Vector projection is introduced for isolating different dual faults. In this method,

the PCA is used for fault signature collection from the samples corresponding to the different single-fault and dual-fault scenarios. Through the clustering techniques described in the previous sections, the first eigenvector of faulty residuals is obtained as the fault direction in the residual space. Finally, the new faulty samples are projected onto each fault direction. If a fault is of a particular type, the projection of the residuals of that fault direction would be higher than those of other directions. In case of a dual-fault scenario occurs, the samples are expected to have large projections on the both single-fault directions in the residual space for the two faults involved. As far as the dual-fault direction is concerned, we expect to see the maximum projection of the faulty sample residuals on the corresponding dual-fault direction.

Figures 7.25 - 7.31 show the plots of the fault directions of the measurements for the first seven dual-fault cases, all of which involve the inlet cooling brine temperature sensor drifting fault. As can be seen, the directional signatures of the studied dual-faults are around 0.2~0.4 when they are projected onto the 8 single-fault directions, as opposed to the much higher values shown in the previous section for the single-fault cases. This is expected due to the difference between single-fault and dual-fault scenarios. Nevertheless, the single-fault directional signatures for the dual-faults are still distinct enough to provide an initial idea of fault isolation. The bottom plots of Figure 7.25-7.31, which are the plots of the residual projections onto the 28 dual-fault directions, clearly illustrate the maximum projection of around one on the corresponding fault direction for each of the seven studied dual-fault cases. All fault cases are diagnosed correctly using the developed fault isolation approach. A caveat with the developed PCA fault diagnostic method is the faults considered for the system of interest must be defined beforehand. In the event of undefined faults occurring in the system, one may be able to observe the faults using the fault directional approach, but the clear fault isolation may not be achievable with the existing fault directions. These faults, if properly identified later, may be included in the fault set.

The fault isolation results for the remaining dual-fault cases are given in Appendix C.

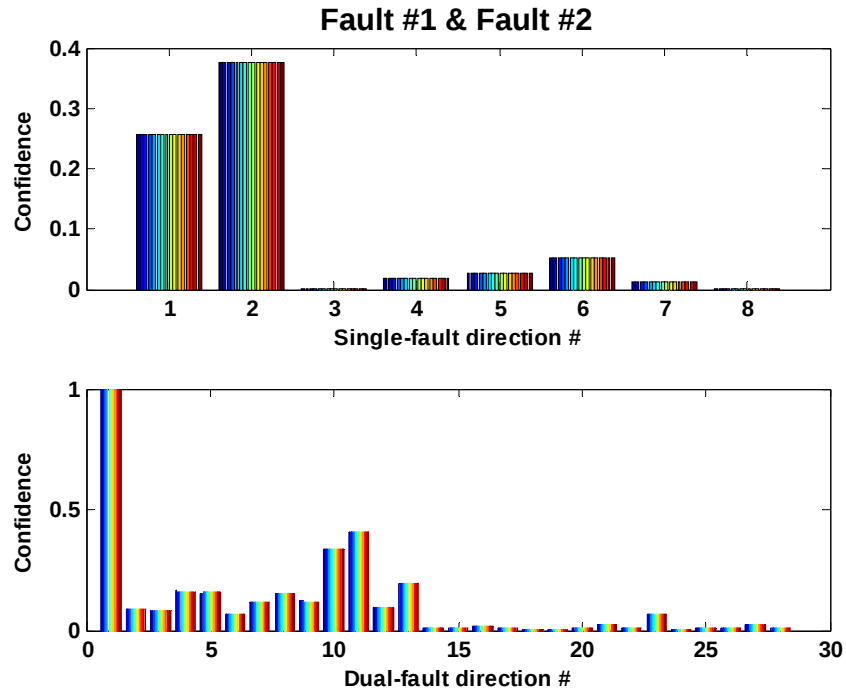


Figure 7.25 Fault isolation index for dual-fault case #1.

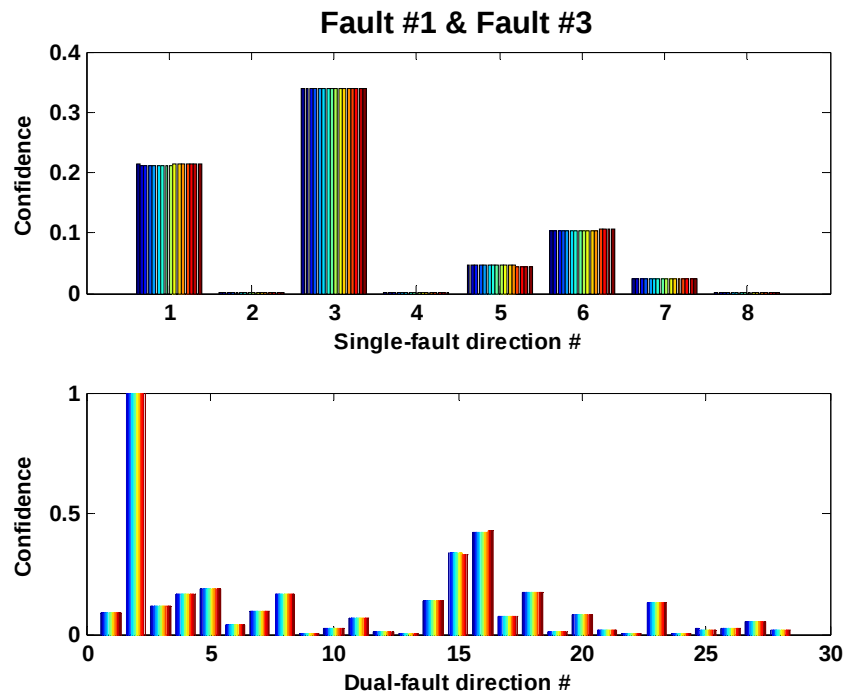


Figure 7.26 Fault isolation index for dual-fault case #2.

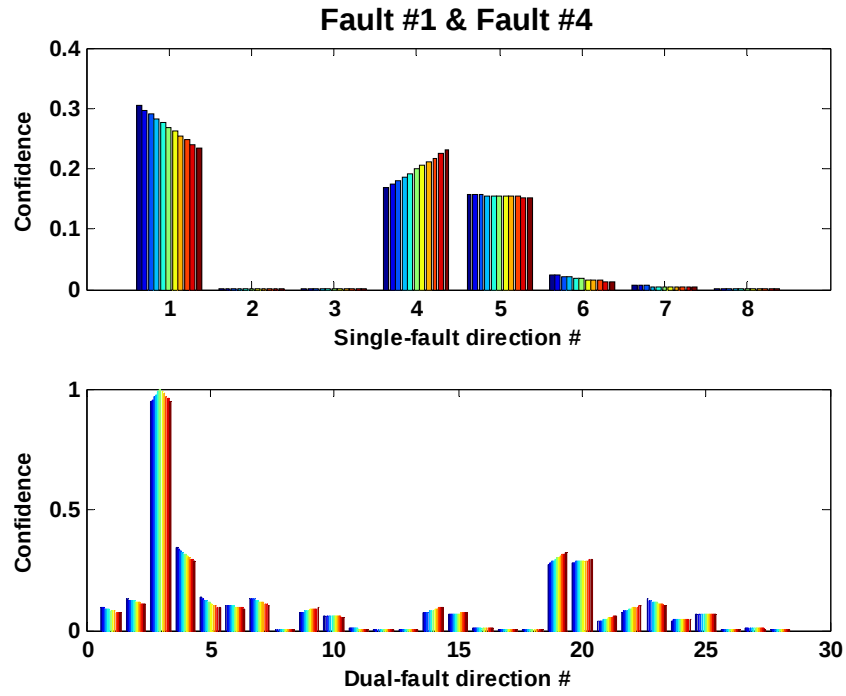


Figure 7.27 Fault isolation index for dual-fault case #3.

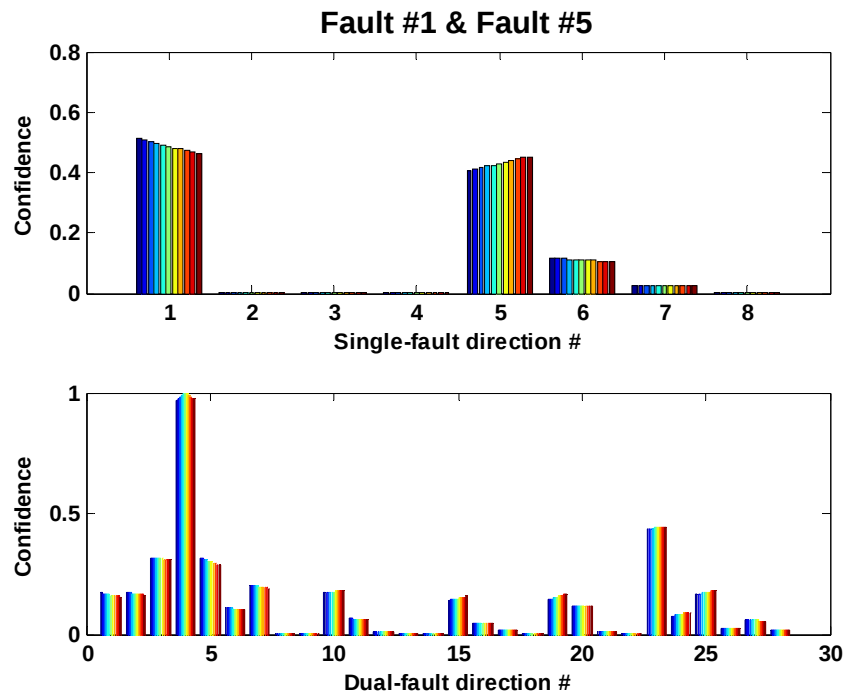


Figure 7.28 Fault isolation index for dual-fault case #4.

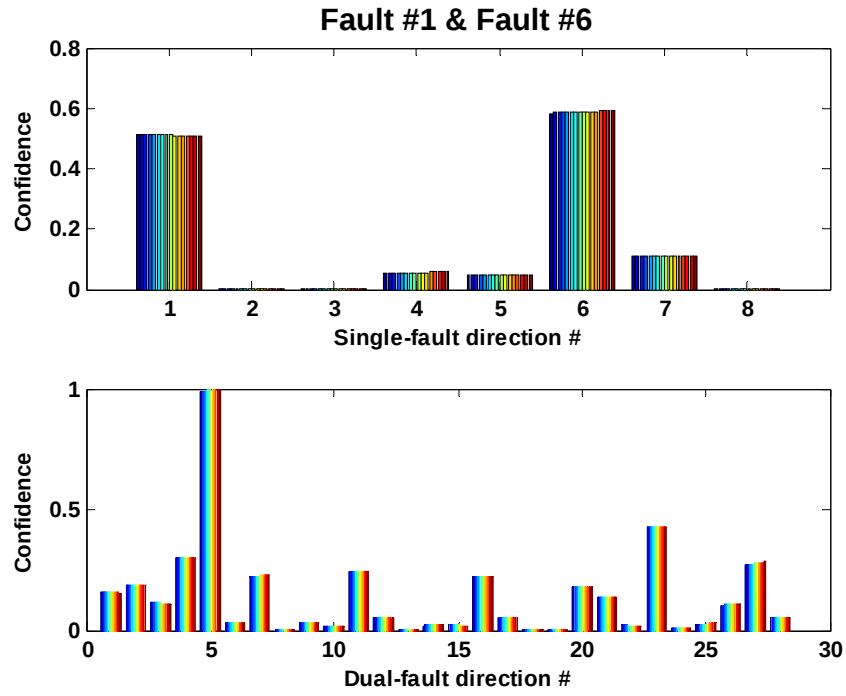


Figure 7.29 Fault isolation index for dual-fault case #5.

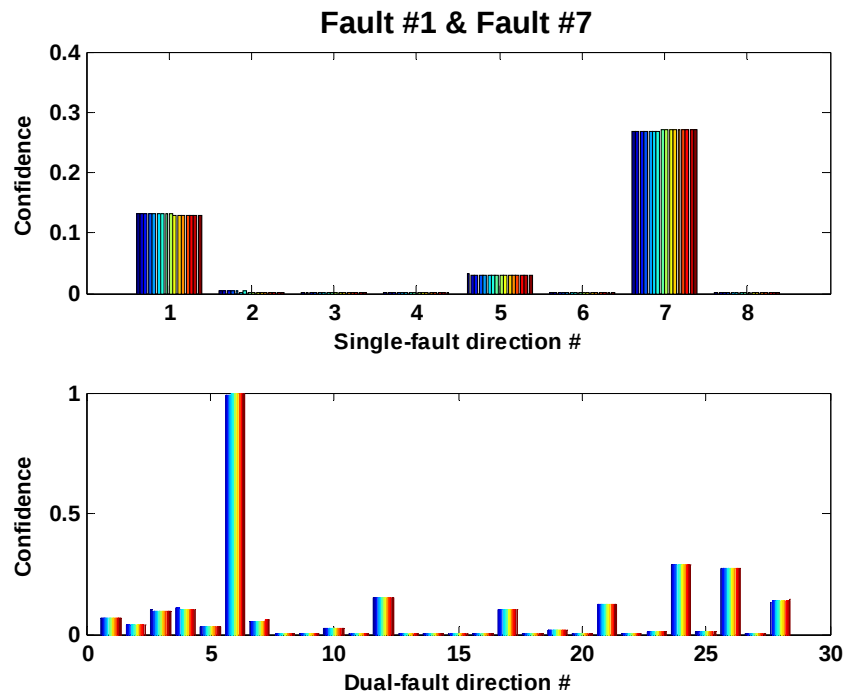


Figure 7.30 Fault isolation index for dual-fault case #6.

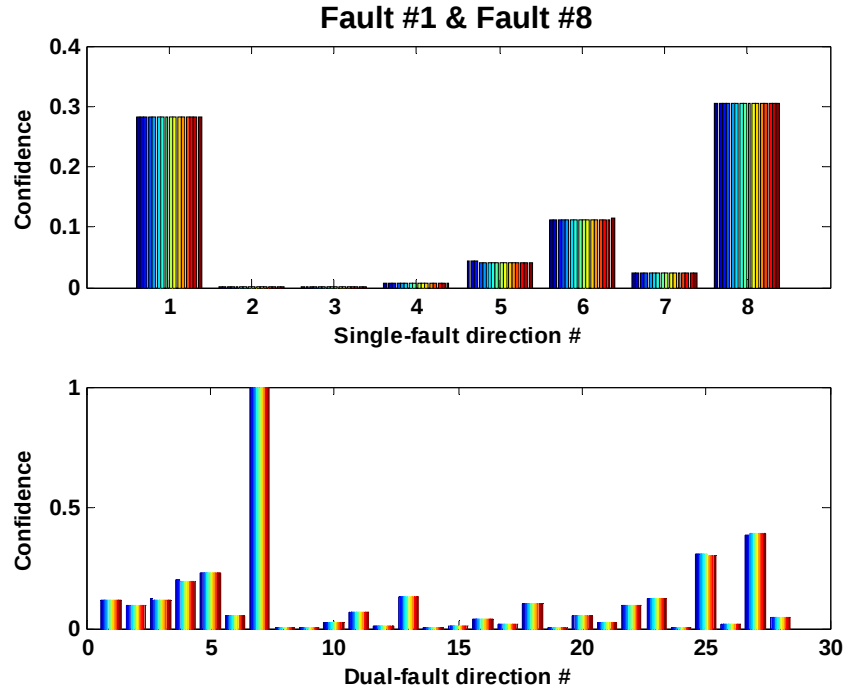


Figure 7.31 Fault isolation index for dual-fault case #7.

7.8 Summary

In this chapter, the developed optimum sensor placement design formulations and a PCA-based fault diagnostic method were employed to diagnose the faults defined within a MSF system. When the optimal sensor sets were used, both sensor faults and process faults were correctly detected and isolated using T^2 and Q statistics, as well as fault isolation index. It was also discovered through the FDI case studies that the PCA fault diagnostic approach could be extended to detect and isolate simultaneous dual-faults using the optimal sensor sets identified for the single-fault cases, in which case both single-fault and dual-fault directions shall be used as fault signatures in order to achieve dual-fault isolation. The fault diagnostic results demonstrated the effectiveness of the developed FDI methods, when used in conjunction with the optimal sensor selection strategy.

Chapter 8

Sensor Placement Design and Fault Diagnosis Application for HCSG Systems

8.1 HCSG System Description

International Reactor Innovative and Secure (IRIS) is one of the next generation nuclear reactor designs for near term deployment. This integral LWR type system has eight helical coil steam generators (HCSG) connected to four steam lines and four feed water lines. Eight steam generators are installed in four pairs in the annular space between the core barrel and the reactor vessel wall, which is shown in Figure 8.1 [Carelli et al., 2003]. On the primary side, each reactor coolant pump (RCP) is dedicated to discharging primary coolant into one steam generator. Therefore, each RCP+HCSG module constitutes a separate flow path. On the secondary side, a common feed water supply line splits at the vessel and goes to two steam generators. Similarly, the steam discharge lines from two steam generators join to create a common steam line.

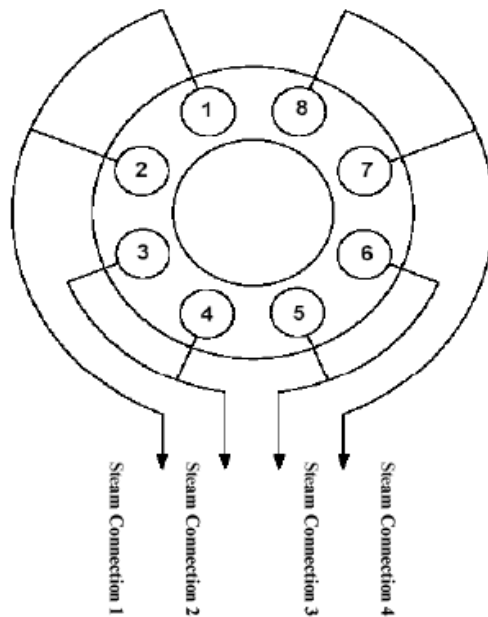


Figure 8.1. IRIS steam generator layout [Carelli et al., 2003].

This pairing design of steam generators reduces the number of feed water and steam lines, and the number of penetrations into and out of the containment, but it has an unfavorable consequence on individual steam generator monitoring in terms of its thermal performance and the secondary flow rate. Figure 8.2 shows a schematic of one pair of steam generators.

Twelve potential sensor locations shown in Figure 8.2 are denoted in numerical order as follows:

1. Primary pump outlet;
2. Primary inlet of steam generator 1;
3. Primary outlet of steam generator 1;
4. Primary inlet of steam generator 2;
5. Primary outlet of steam generator 2;
6. Cold leg connection line;
7. Feedwater connection line;
8. Secondary inlet of steam generator 2;
9. Secondary outlet of steam generator 2;
10. Secondary inlet of steam generator 1;
11. Secondary outlet of steam generator 1;
12. Steam connection line.

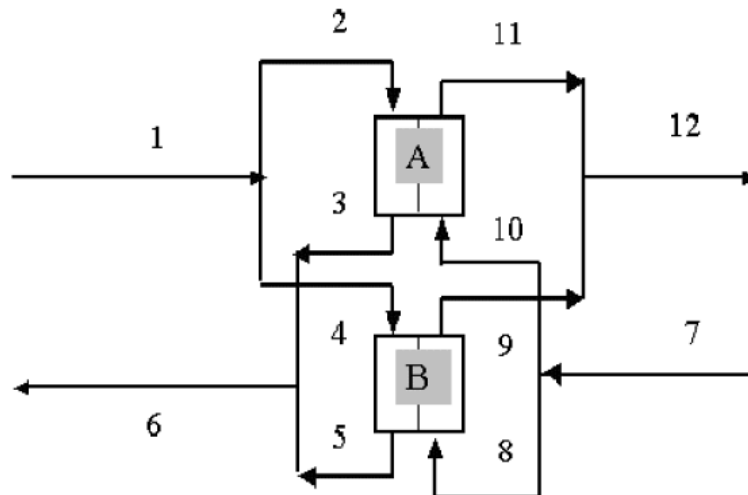


Figure 8.2. A schematic of a pair of HCSGs [Zhao, 2005a].

For one pair of HCSGs, the steady state models that are based on mass balance and heat balance equations are given as follows.

$$W_1 - W_2 - W_4 = 0$$

$$W_2 - W_3 = 0$$

$$W_4 - W_5 = 0$$

$$W_6 - W_3 - W_5 = 0$$

$$W_{12} - W_9 - W_{11} = 0$$

$$W_{10} - W_{11} = 0$$

$$W_8 - W_9 = 0$$

$$W_7 - W_8 - W_{10} = 0$$

$$T_1 = T_2 = T_4$$

$$T_7 = T_8 = T_{10}$$

$$W_6 h_6 - W_3 h_3 - W_5 h_5 = 0$$

$$W_{12} h_{12} - W_9 h_9 - W_{11} h_{11} = 0$$

$$W_2 h_2 + W_{10} h_{10} - W_3 h_3 - W_{11} h_{11} = 0$$

$$W_4 h_4 + W_8 h_8 - W_9 h_9 - W_5 h_5 = 0$$

The symbols $W, h, \text{ and } T$ denote mass flow rate, specific enthalpy, and temperature, respectively. The numeric number appended to the symbols corresponds to a potential sensor location in the HCSG system configuration, as shown in Figure 8.2. It is assumed that the system pressures are constant. Thus, the specific enthalpy can be approximated as a function of local fluid temperature. We shall build a HCSG digraph upon these steady state linear balance equations for one pair of HCSGs due to the symmetry of their configuration.

The HCSG digraph is shown in Figure 8.3. The yellow nodes represent the root nodes of the system, each of which corresponds to a fault. Table 8.1 lists the six faults that the HCSG diagnostic system needs to monitor. Both process faults and sensor faults are considered in the HCSG fault diagnosis. The three sensor faults considered are sensor drifts; and the thermal degradation of SG-A and SG-B, as well as the secondary flow distribution anomaly, are process faults considered for the HCSG systems. The digraph has clearly illustrated the cause-effect relationships among the involved variables and the propagation pathways from the fault nodes to the other nodes.



Fault Nodes	System Faults
F1	Hot leg temperature sensor fault
F2	SG-A heat transfer degradation
F3	SG-B heat transfer degradation
F4	Secondary flow distribution anomaly
F5	Feedwater temperature sensor fault
F6	Feedwater flow sensor fault

8.2 Sensor Placement Design

8.2.1 Fault Diagnostic Observability Criterion

As a first step, the greedy search algorithm for fault observability is applied to find the minimum set of sensors required to observe all the faults listed in Table 8.1 for one pair of HCSGs. This gives nodes $[T_3, T_5]$ as the sensor set. It is noted that in this case, nodes can also be the sensor set, because they are connected to all the faults. Although all the faults can be detected, every one of them, however, cannot be distinguished from one another.

The algorithm is applied to obtain the set of sensors that will give maximum resolution under the single-fault assumption case, which generates $[T_3, T_5, T_9, T_{11}, W_7, W_{12}]$ as the minimum sensor set for fully isolation of the selected faults. Note that none of the fault nodes affect the same sensor set, therefore, they should be distinguished from one another by the selected sensor set.

8.2.2 System Unobservability Criterion

In this sub-section, sensor placement design is carried out for the pair of HCSGs by solving the optimization problem that minimizes the system overall unobservability, which is defined as the maximum unobservability across all faults. The objective function of the optimization is given in Equation (6.13). The cost is assumed to have the same value at 100 per sensor for all the sensors. The fault occurrence probabilities are listed in Table 8.2. The sensor failure probabilities are given in Table 8.3. The previously developed greedy search heuristic is employed to solve for optimal selection of the sensors. For the sake of comparison, the one-step optimization problem formulated in Equation (6.18) is solved by the LINGO 8.0 optimization software package under the same cost constraints as greedy search. The one-step approach minimizes the system overall unobservability and, among candidates offering the minimum system unobservability, selects the one with the least cost.

Table 8.4 shows the sensor selection results under a variety of total available cost constraints. The objective function, that is, system overall unobservability, is also reported in this table. It is seen that the greedy search heuristic offers the results similar to those obtained by LINGO 8.0. And the main difference between the two distinct optimization algorithms is that the greedy search tends to use all the available resources, while LINGO sometimes achieves the same system unobservability with fewer sensors.

Table 8.2. Fault data for a pair of HCSGs

Fault Nodes	Occurrence Probability ($\log f_j$)	Fault Nodes	Occurrence Probability ($\log f_j$)
F1	-2	F4	-1
F2	-2	F5	-1
F3	-2	F6	-1

Table 8.3. Sensor data for a pair of HCSGs

Sensor Nodes	Variable	Failure Probability ($\log s_i$)	Sensor Nodes	Variable	Failure Probability ($\log s_i$)
S1	T_1	-2	S12	W_4	-1
S2	T_2	-2	S13	W_7	-1
S3	T_3	-2	S14	W_8	-1
S4	T_4	-2	S15	W_{10}	-1
S5	T_5	-2	S16	W_{12}	-1
S6	T_7	-2	S17	T_6	-2
S7	T_8	-2	S18	T_{12}	-2
S8	T_9	-2	S19	W_3	-1
S9	T_{10}	-2	S20	W_5	-1
S10	T_{11}	-2	S21	W_{11}	-1
S11	W_2	-1	S22	W_9	-1

Table 8.4. Sensor selection results using greedy search heuristic and LINGO 8.0

Total Cost	System Unobservability	Sensors Selected	
		LINGO 8.0	Greedy Search
400	0.01	S3, S5, S6, S13	S3, S5, S6, S13
500	0.01	S3, S5, S6, S13	S3, S5, S6, S13 ₍₂₎
800	0.001	S3, S5, S6, S11, S12, S13 ₍₂₎ , S14	S3, S5, S6, S11, S12, S13 ₍₂₎ , S14
1000	0.001	S3, S5, S6, S11, S12, S13 ₍₂₎ , S14	S1, S3, S5 ₍₂₎ , S6, S11, S12, S13 ₍₃₎
1500	10 ⁻⁴	S1, S3, S5 ₍₂₎ , S6, S11 ₍₂₎ , S12 ₍₂₎ , S13 ₍₃₎ , S14	S1, S3 ₍₂₎ , S5 ₍₂₎ , S6 ₍₂₎ , S11 ₍₂₎ , S12 ₍₂₎ , S13 ₍₄₎
2000	10 ⁻⁵	S1, S3 ₍₂₎ , S5 ₍₂₎ , S6 ₍₂₎ , S11 ₍₃₎ , S12 ₍₃₎ , S13 ₍₄₎ , S14	S1 ₍₂₎ , S3 ₍₂₎ , S5 ₍₂₎ , S6 ₍₂₎ , S11 ₍₄₎ , S12 ₍₄₎ , S13 ₍₄₎

In order to show the decreasing trend of the system unobservability as the iteration progresses, the greedy search algorithm is applied to the HCSG system when the total available cost is 1000. The results are tabulated in Table 8.5. The first column in the table indicates the number of available sensors. For each extra available sensor, the positions of the already located sensors do not change. Hence only the additional sensor is listed in the second column of Table 8.5. The third column in the table is the unobservability of the fault with the highest value after sensor selection. The sensor location results are shown for ten available sensors.

In order to verify the effectiveness of the greedy search optimization algorithm, the unobservability values of the six faults are calculated. When the total available cost is 1500, it takes 16 iterations before the algorithm stops. Figure 8.4 shows the trending of unobservability. It can be clearly seen that the greedy search optimization algorithm reduces the unobservability values for all the six faults.

Table 8.5. HCSG sensor placement results ($TC=1000$)

# of available sensors	Additional sensor	System unobservability after sensor selection
1	S5	0.1
2	S3	0.1
3	S6	0.1
4	S13	0.1
5	S13	0.01
6	S1	0.01
7	S11	0.01
8	S12	0.01
9	S13	0.001
10	S5	0.001

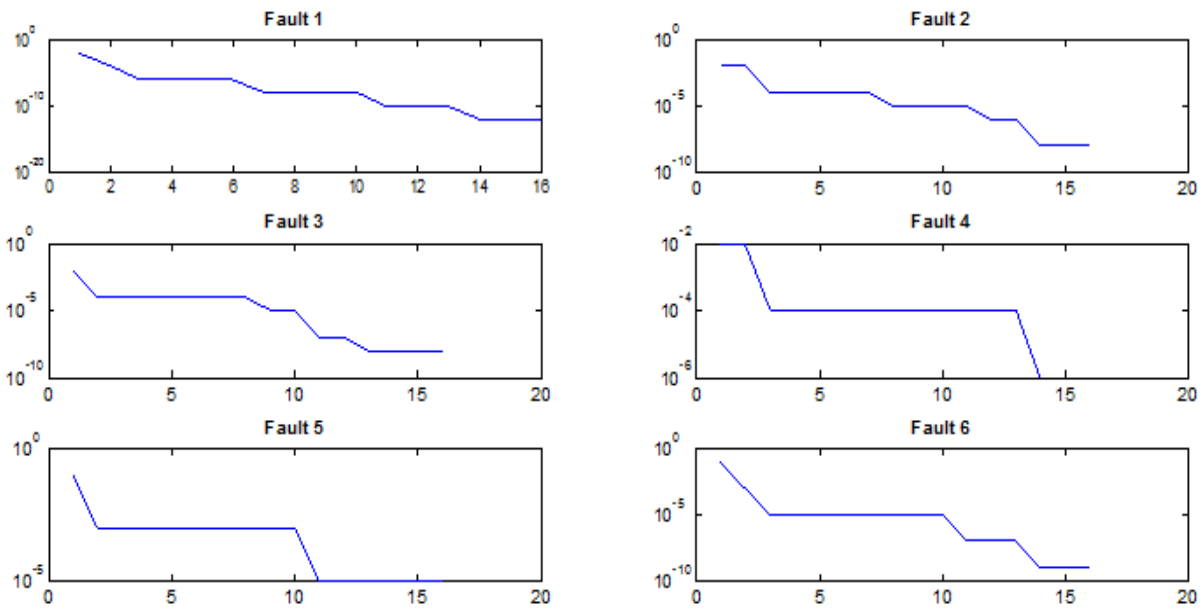


Figure 8.4. Unobservability trending of the faults in a pair of HCSGs.

8.3 PCA Fault Diagnostic Results

For a pair of HCSGs, the digraph has been constructed based on mass and heat balance equations, as shown in Figure 8.3. In the previous section, the design problem of sensor network with maximum fault resolution using fault observability criterion has been studied. The greedy search algorithm gives $[T_3, T_5, T_9, T_{11}, W_7, W_{12}]$ as the minimum sensor set for fully isolation of the selected faults. The advantage of using this optimized sensor set is that more information about the system is utilized, and some basic properties such as the fault detectability and identifiability are already partially guaranteed before PCA has been used to monitor system behavior.

To build the model for nominal operation, a simulation database was created for different power levels ranging from 40% to 100% of the plant capacity. The data were generated for one pair of the steam generators, SG-A and SG-B, to simulate the measurements in actual nuclear power plants. For this pair of steam generators, the above six sensors identified as the optimized sensor set are used as the measured variables to provide process information of the HCSG systems. (a) T3: cold leg temperature of SG-A; (b) T5: cold leg temperature of SG-B; (c) T9: steam temperature leaving the secondary side of SG-B; (d) T11: steam temperature leaving the secondary side of SG-A; (e) W7: feed water flow rate into the secondary side of SG-A and SG-B; (f) W12: steam flow rate leaving the secondary side of SG-A and SG-B.

A PCA model was developed using the data for the nominal operation case with the six variables in the minimum sensor set. The nominal operation data matrix is preprocessed by auto-scaling the columns in the data matrix to zero mean and unit variance. This puts all the measurements with their different units on a common unit variance scale. Under normal conditions, small residuals would be generated and limited to a certain range. In the test cases when one or several components in the system are under degradation, the causal relations among the variables will be violated. As a result, the mapping of residuals from residual generators or system models will increase in a specific direction. The residual patterns for different types of faults are shown in Figure 8.5-8.10.

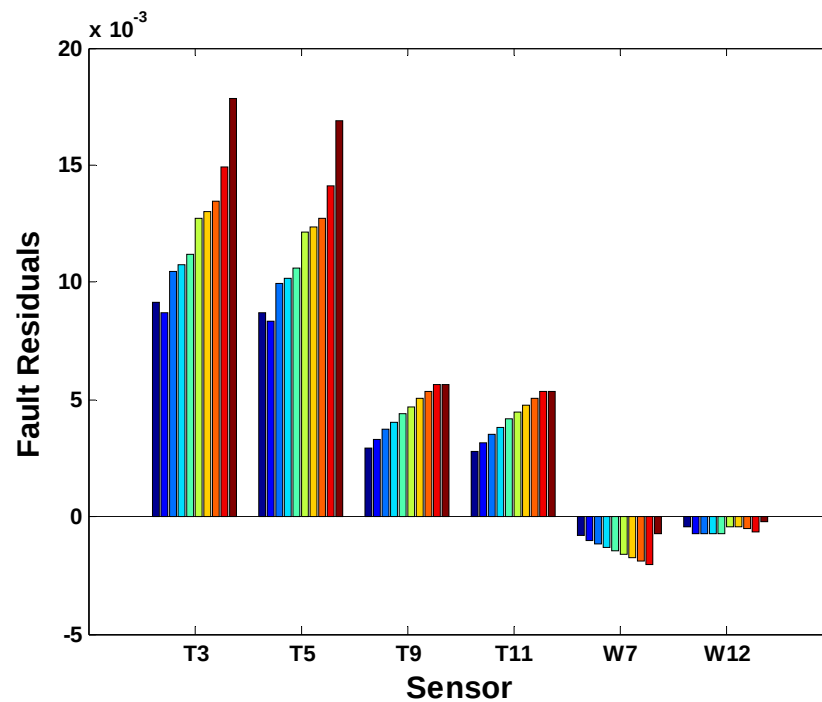


Figure 8.5. Residual pattern for T_{hot} sensor fault (Fault #1).

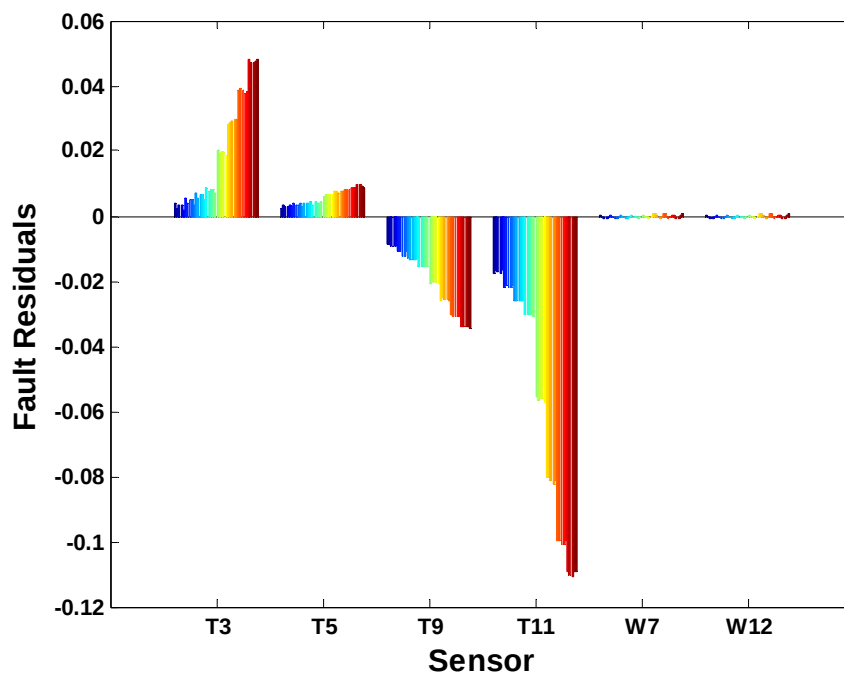


Figure 8.6. Residual pattern for SG-A thermal degradation (Fault #2).

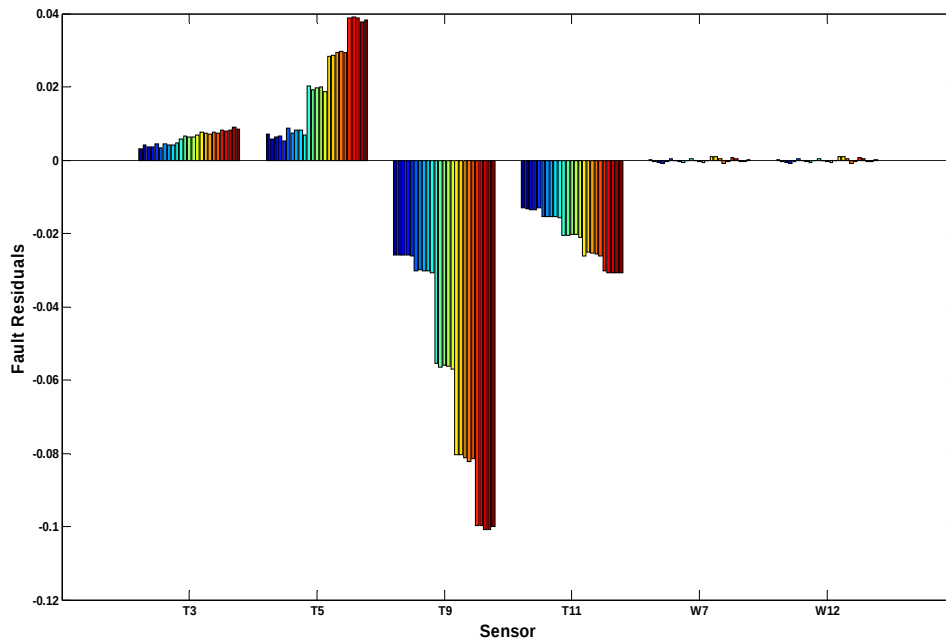


Figure 8.7. Residual pattern for SG-B thermal degradation (Fault #3).

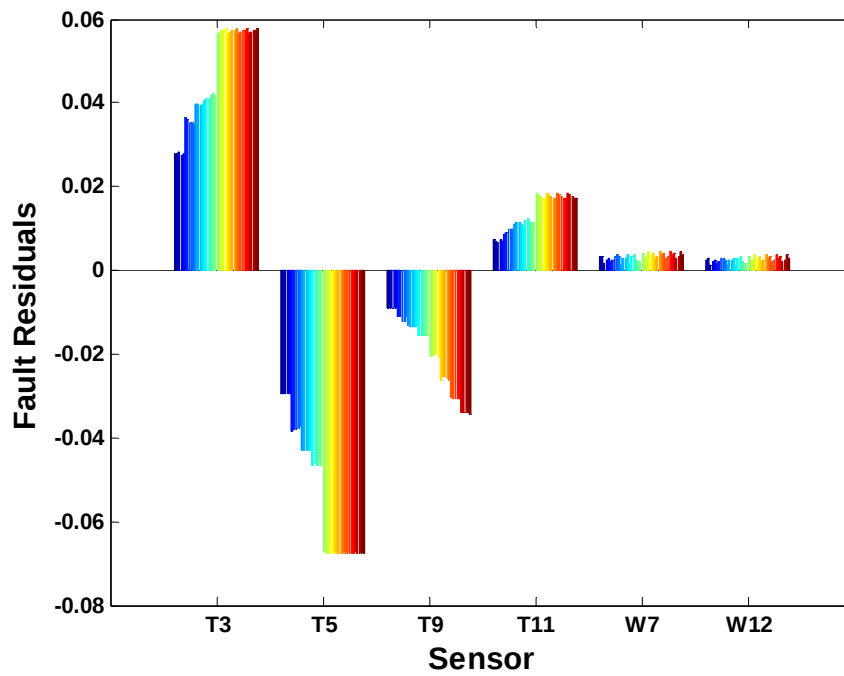


Figure 8.8. Residual pattern for secondary flow distribution anomaly (Fault #4).

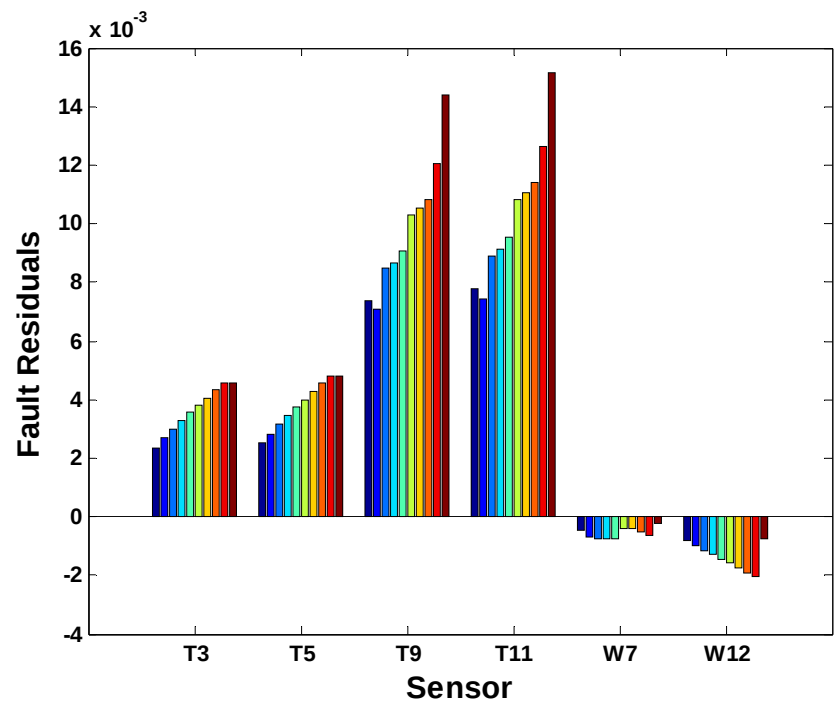


Figure 8.9. Residual pattern for feed temperature sensor fault (Fault #5).

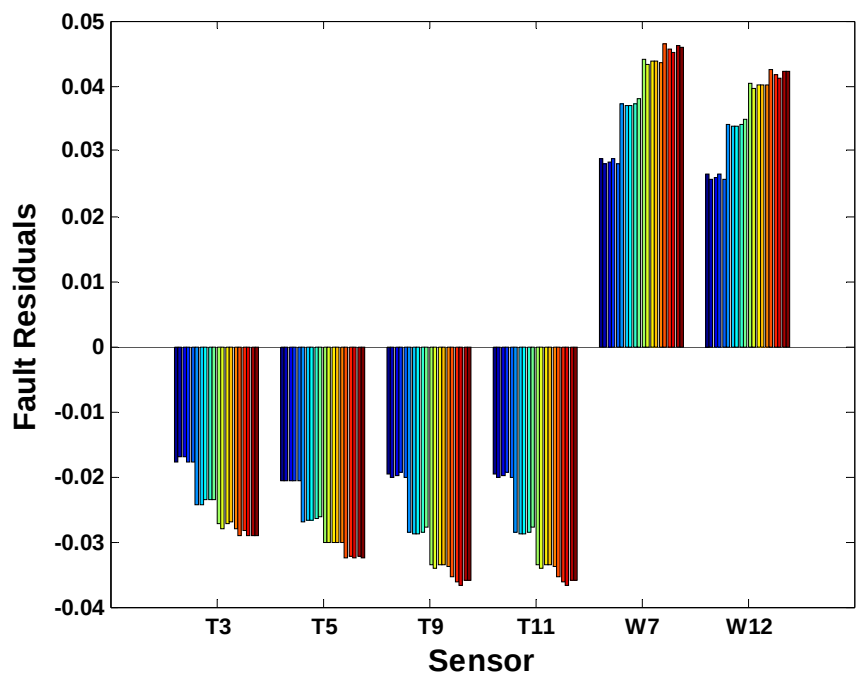


Figure 8.10. Residual pattern for feed flow sensor fault (Fault #6).

To understand the fault residual patterns, the residual pattern plot for the secondary flow distribution anomaly (Fault #4), shown in Figure 8.8, is discussed here. The secondary flow distribution anomaly is a process fault in the HCSG systems. When this process fault occurs, the flow rate into the secondary side of each steam generator will be different. However, because the secondary fluid flows inside the helical coil tubes, it is unrealistic to directly measure the flow rate into each steam generator and the fault effects cannot be directly observed based on the flow rates. For this reason, the fault needs to be monitored from the other measured variables such as the primary outlet temperatures and the steam outlet temperatures.

The fault data for the secondary flow distribution anomaly (Fault #4) is generated by linearly reducing the feed flow rate into SG-A from 100% value to 90% value, while increasing the feed flow rate into SG-B from 100% to 110% value. Thus as shown in Figure 8.8, the temperatures T3 and T11 have positive components, indicating the increase in both measurements when the secondary flow rate into the SG-A decreases; and the temperatures T5 and T9 have negative components, indicating the decrease in them when the secondary flow rate into the SG-B increases.

Then, the PCA fault direction analysis is performed for processing the fault residuals. The six fault directions which correspond to the six fault scenarios are listed in Table 8.6. The following plots (Figures 8.11-8.16) illustrate the fault isolation indices for the different fault cases. Note that the fault index is close to unity for the particular fault that occurs and is smaller for the others. It is shown that all the fault cases are distinguished correctly using the fault isolation approach when the PCA model includes the six sensors that are chosen by the sensor placement algorithm.

Table 8.6. PCA fault direction analysis for a pair of HCSGs

Fault direction #	System faults	% of residual variances explained by the 1st PC
1	Hot leg temperature sensor fault	98.22
2	SG-A thermal degradation	96.71
3	SG-B thermal degradation	94.52
4	Secondary flow distribution anomaly	92.06
5	Feed temperature sensor fault	95.84
6	Feed flow sensor fault	94.39

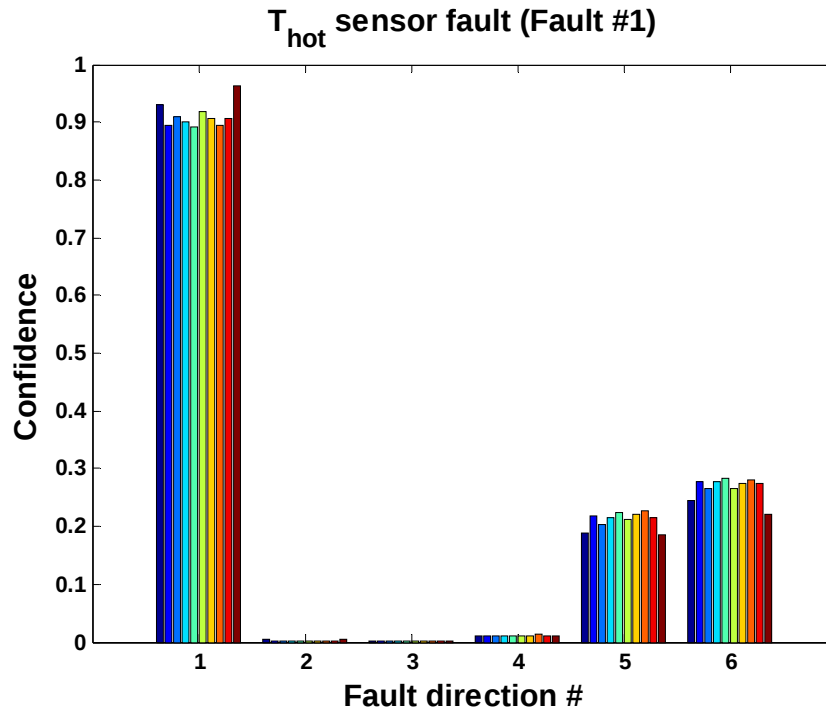


Figure 8.11. Fault isolation index for T_{hot} sensor fault.

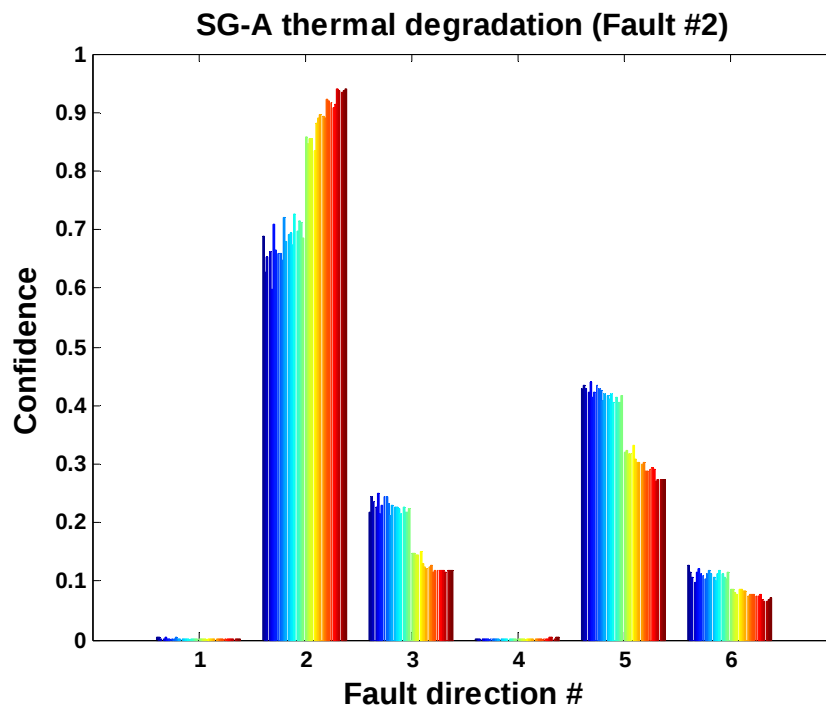


Figure 8.12. Fault isolation index for SG-A thermal degradation.

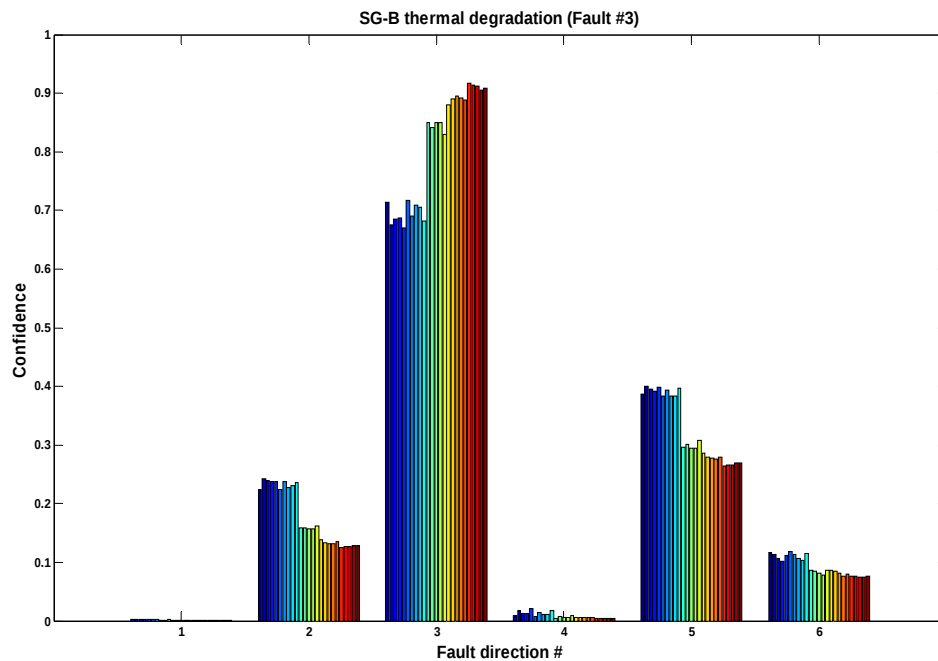


Figure 8.13. Fault isolation index for SG-B thermal degradation.

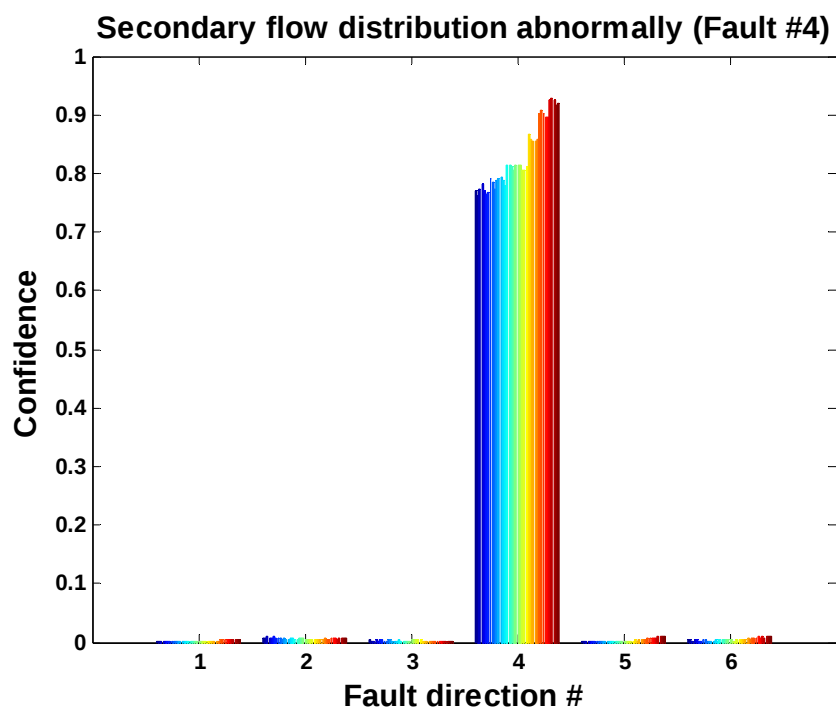


Figure 8.14. Fault isolation index for secondary flow distribution anomaly.

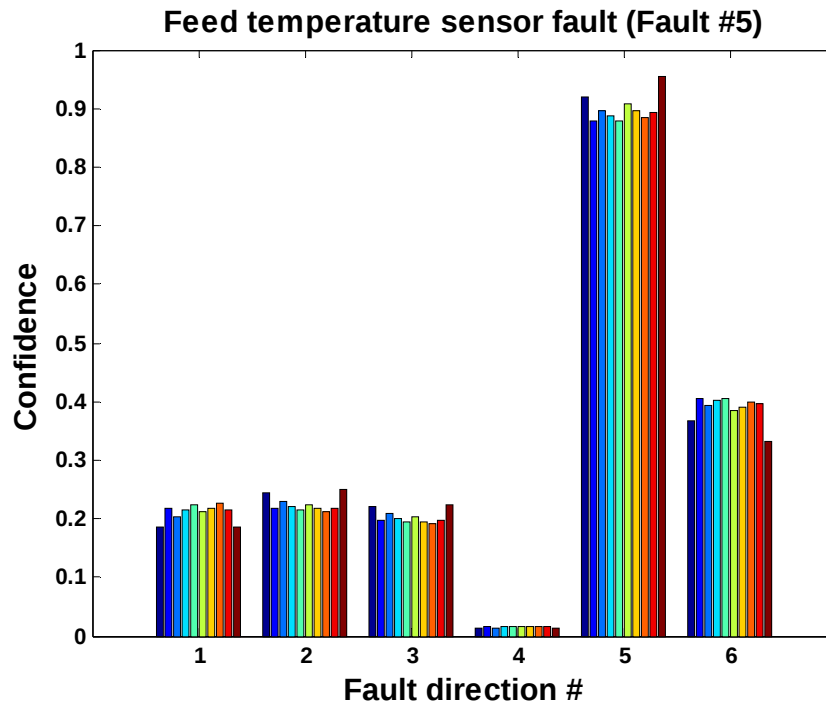


Figure 8.15. Fault isolation index for feed temperature sensor fault.

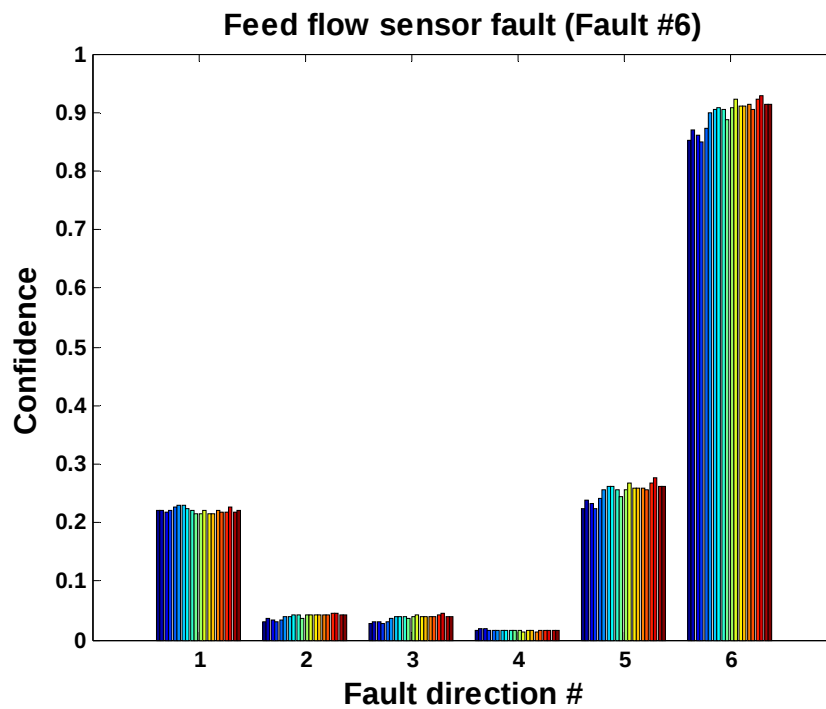


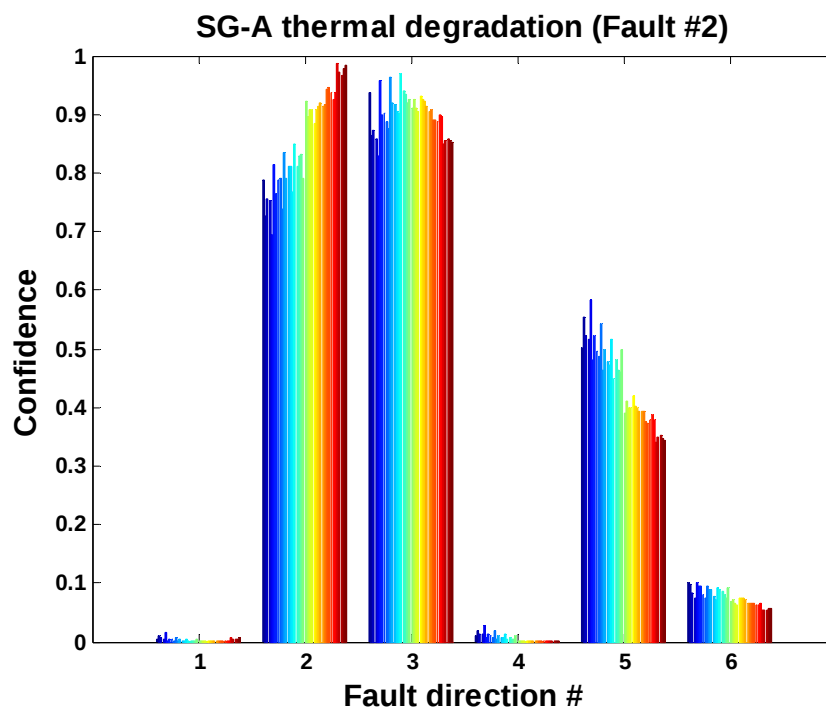
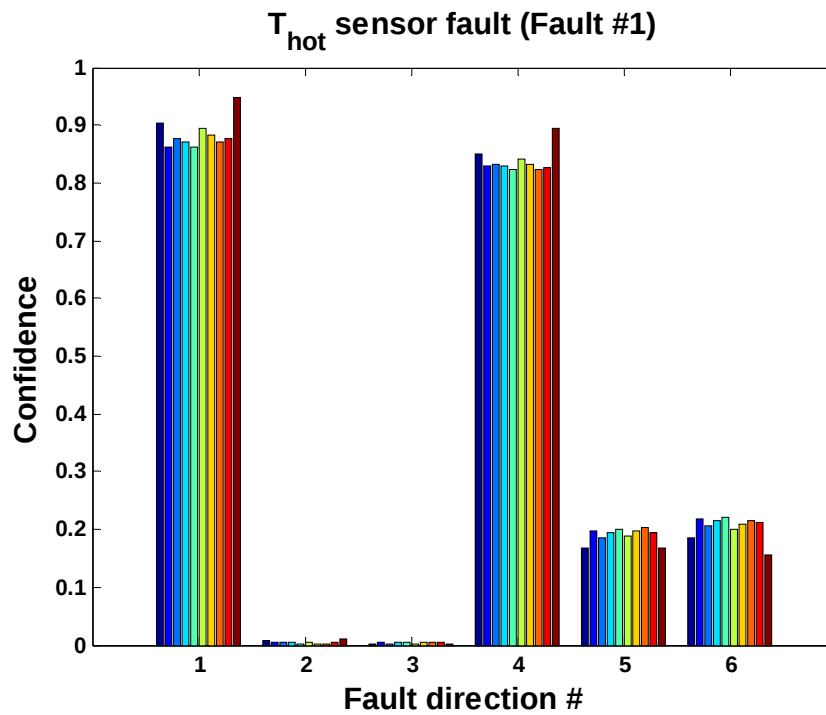
Figure 8.16. Fault isolation index for feed flow sensor fault.

For the sake of comparison, the following plots (Figures 8.17-8.22) illustrate the fault isolation indices for the different fault cases when the reduced sensor set is used to develop the PCA model for HCSG fault monitoring. For instance, only T3, T11, W7, and W12 are included in this case study.

In summary, Figure 8.23(a) shows the fault isolation indices for each of the detected fault scenarios in the HCSG system. All the fault cases are detected correctly using the fault isolation approach. In comparison, Figure 8.23(b) illustrates the fault isolation indices when the reduced sensor set (T3, T11, W7, and W12) is used to develop the PCA model for HCSG fault monitoring. It is seen that when the hot leg temperature sensor is under degradation (Fault #1), the fault isolation indices of both fault direction #1 and fault direction #4 have significant values, which are difficult to distinguish. The same problems occur for the fault cases 2, 3, and 4. Therefore, we can draw the conclusion that the reduced sensor set is not sufficient for the purpose of fault isolation in the HCSG systems, thus illustrating the effectiveness of optimal sensor allocation strategy.

8.4 Summary

In this chapter, the application of optimal sensor allocation for the IRIS helical coil steam generator system is demonstrated. A PCA-based fault isolation method is introduced to generate fault signatures of typical faults found in the HCSG systems. The confidence levels on each fault direction were calculated through vector projection. Fault isolation index provides a convenient way to isolate the faults using projections on the fault directions. Fault diagnostic results were also examined using reduced sensor sets (non-optimal sensor selection) for the same fault scenarios, and the results indicated that the reduced sensor sets were not sufficient for the purpose of fault isolation.



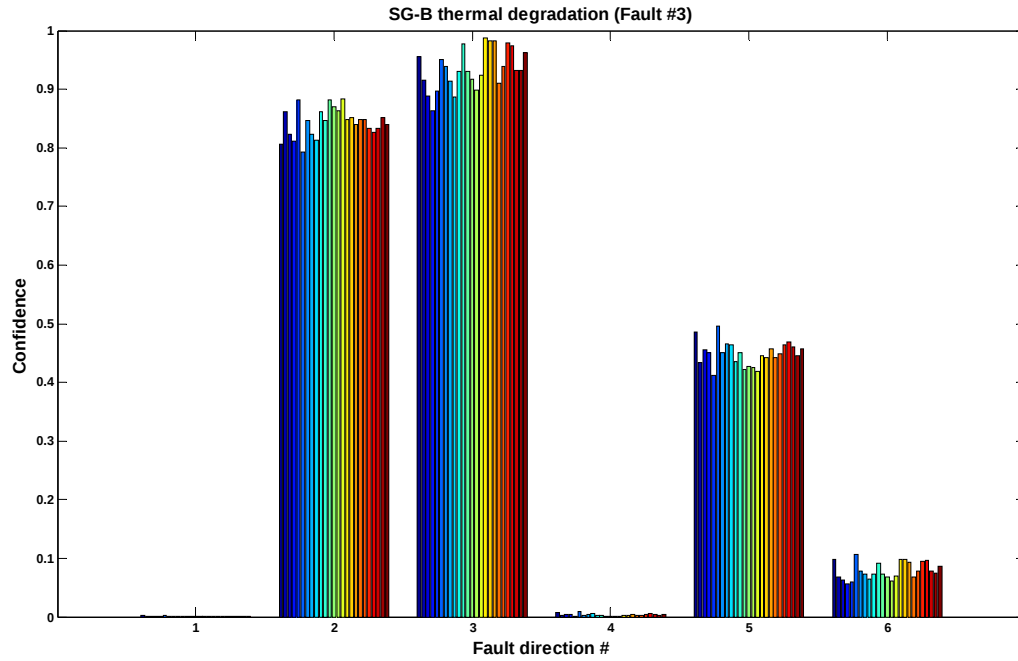


Figure 8.19. Fault isolation index for SG-B thermal degradation (reduced sensor set).

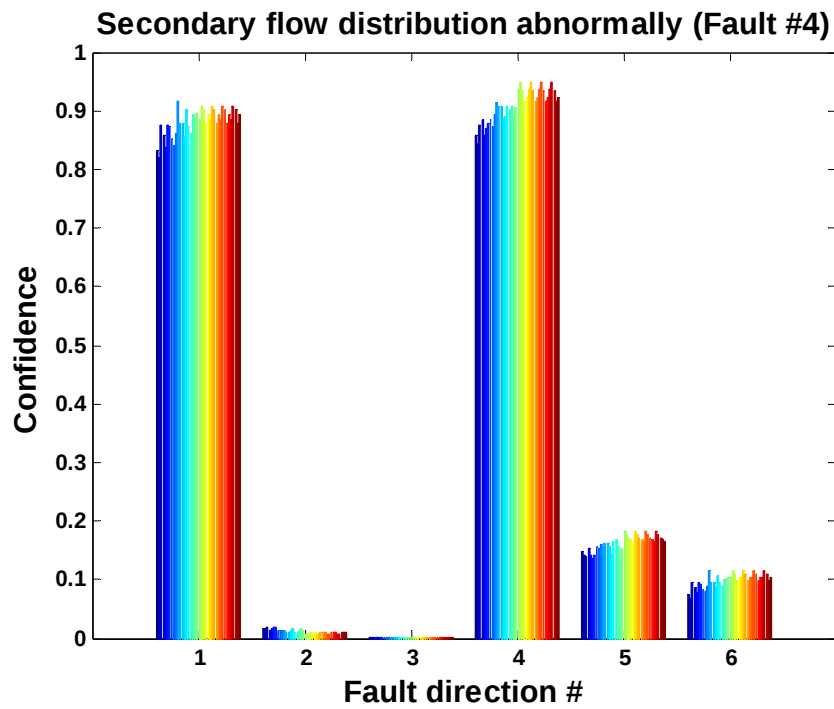


Figure 8.20. Fault isolation index for secondary flow anomaly (reduced sensor set).

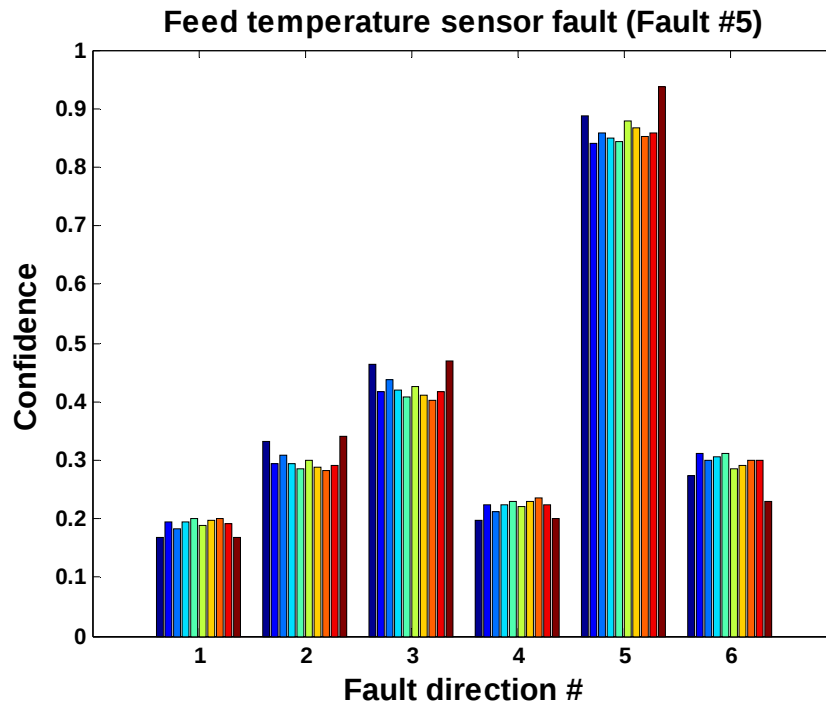


Figure 8.21. Fault isolation index for feed temperature sensor fault (reduced sensor set).

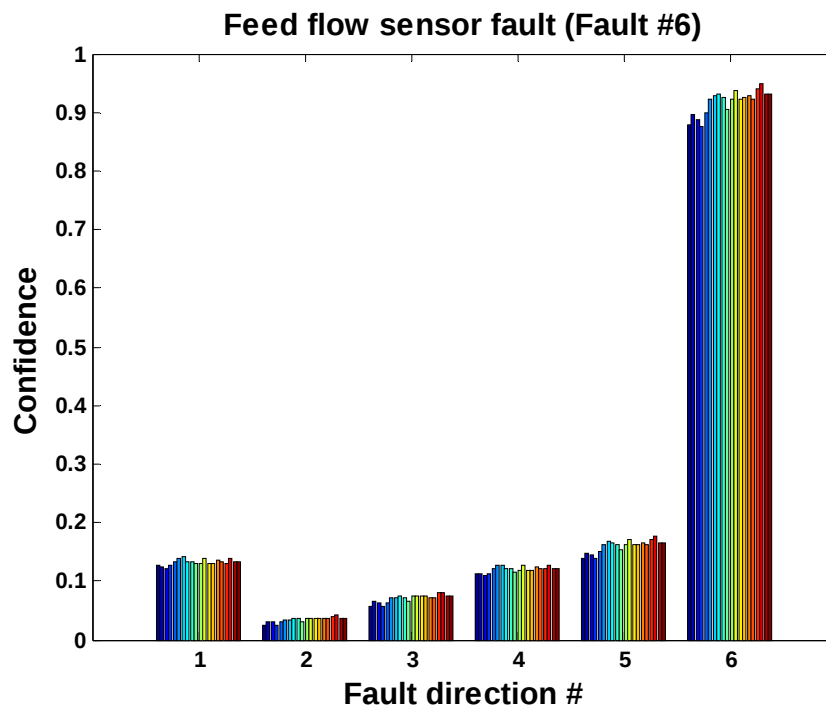


Figure 8.22. Fault isolation index for feed flow sensor fault (reduced sensor set).

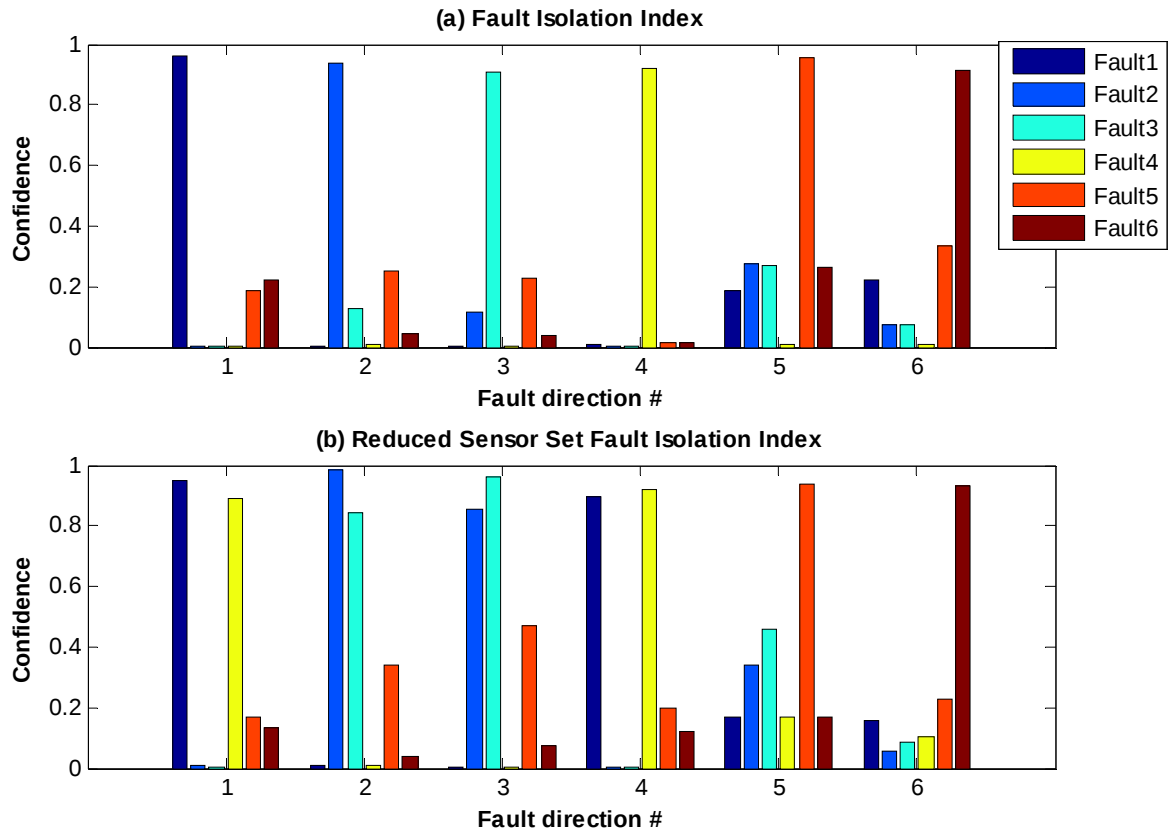


Figure 8.23. Fault isolation index for a pair of HCSGs.

Chapter 9

Summary, Conclusions, and Suggestions for Future Research

9.1 Summary and Conclusions

Both nuclear and seawater desalination are mature and proven technologies, and are commercially available from a variety of suppliers. Nuclear plants can provide both electrical and thermal energy in an integrated, co-generation fashion to produce a spectrum of energy products including electricity, process heat, district heating, and potentially hydrogen generation. The prospects of using nuclear energy for seawater desalination on a large scale are very attractive since desalination is an energy-intensive process that can utilize the heat and/or the electricity produced by nuclear plants. While all nuclear reactor types can provide the energy required by the various desalination processes, it has been shown that small and medium modular reactors, a burgeoning field in the nuclear community today, offer the largest potential as coupling options of nuclear desalination systems in some developing countries where less stable electrical grids and increasing necessity for potable water sources are often found. IRIS is such a novel SMR with an advanced, modular, passively safe design that is well suited for desalination applications. However, coupling a SMR system and a desalination process involves a number of issues that have to be addressed in the design phase. Among these issues, instrumentation and controls, performance monitoring and fault diagnosis are of high importance for the safe and optimal operation of a coupled nuclear desalination plant.

The preceding chapters have presented an integrated approach to the performance monitoring and fault diagnosis of a large industrial plant, such as nuclear desalination systems using robust data-based empirical methods. Sensor placement design is a critical piece in a fault monitoring system, which typically includes process performance monitoring, fault detection and isolation, and maintenance planning. Thus, it is emphasized in the developed integrated approach. Nowadays, most processes are extremely well instrumented with a multitude of sensors providing both control and safety-related measurements, especially in a nuclear desalination plant. The ability of the sensor network to detect and discriminate failure modes and anomalous

conditions is critical for the effectiveness of any fault monitoring system. With hundreds of process variables available for measurement in both nuclear power plants and desalination processes, selection of optimum number of sensors and their locations poses a unique challenge.

Six areas of original contribution are discussed in the dissertation. These contributions are innovative and significant in that, to the best of the author's knowledge, they represent the first comprehensive studies of modeling, simulation, sensor placement design, and fault diagnosis of a nuclear desalination process. The original contributions are summarized here.

1. Both steady-state and dynamic models were developed for MSF desalination systems using MATLAB and SIMULINK. The steady-state model was simulated for a real plant operating conditions and a close agreement was observed between the predicted results and the actual plant operating parameters. Dynamic models were suitable for solving problems involving transient behavior, such as studying control strategies, stability problems, process interactions, start-up and shut-down conditions. In this research, the developed MSF dynamic model is a lumped parameter model that consists of ordinary differential equations and supporting algebraic equations. The initial conditions were known from steady-state simulations. The developed dynamic model was used to simulate the effect of step changes in several key state variables. The simulation results obtained were physically plausible and similar to those found in the literature.

2. A dynamic SIMULINK model of the coupled IRIS-MSF nuclear desalination system was also developed in this dissertation research. An existing IRIS SIMULINK model was used for coupling with the developed dynamic MSF model. A simple BOP system was also included in the coupled SIMULINK model. The BOP model incorporated the subsystem models of the turbine, condenser, and feedwater heater systems. The dynamic model of the coupled plant was simulated for a 10% step decrease in the turbine power demand, and the simulation results were examined and analyzed. The results showed that the developed dynamic model was able to characterize system dynamic behavior with reasonably good accuracy to study the control and fault diagnosis of the IRIS-MSF coupled system.

3. An integrated architecture of performance monitoring and fault diagnostics with emphasis on the importance of optimum sensor placement was developed in this dissertation. The solution to the problem of optimum sensor placement from a fault diagnosis perspective can be broadly broken down into two tasks: (1) fault modeling or prediction of cause-effect behavior of the system, generating a set of variables that are affected whenever a fault occurs, and (2) use of the generated sets to identify sensor locations based on various design criteria, such as fault observability, fault resolution, etc. In this dissertation, the DG causal graph modeling method that represents cause-effect behavior of the process was used as a basis for the sensor placement design. Issues such as fault observability and fault resolution are discussed to ensure the designed sensor network could observe every defined fault in the process, meanwhile obtain a maximum possible fault resolution.

4. Development of an optimal and automated sensor placement procedure was conducted as well. The developed sensor placement design algorithm treats finding the optimal sensor locations as one of choosing the minimum number of sensors that would cover all the faults defined for the system of interest, which is the commonly known “minimum set-covering problem”. The fault observability and resolution problems could be solved exactly by enumeration, but with an increasing number of system faults and sensor locations, it may not be computationally feasible to solve the problem in that fashion. In many instances, heuristics often give a quick and reasonably approximate solution. An ILP embedded greedy search heuristic was developed for solving the fault observability and resolution problems in this research. The optimal sensor placement design in terms of fault diagnosis was further formulated as a system unobservability minimization problem which took into account the cost of building a sensor network. Moreover, various information features, such as fault occurrence probability and sensor failure probability, were incorporated in the formulation. The greedy search heuristic was applied to solve the formulated optimization problem, and the optimization performances were evaluated and compared with a commercial optimization solver, LINGO 8.0. Application results showed that the greedy search heuristic and the LINGO solver resulted in optimal sensor sets of comparable performance.

5. A PCA-based fault diagnostic approach, in conjunction with optimized sensor selection, was developed in this dissertation with applications to an IRIS-MSF nuclear desalination plant. Despite using the optimal sensor set obtained from sensor placement design, it was found that some faults of the system were still indistinguishable in many cases. Nonetheless, optimal sensor placement design from a fault diagnosis perspective would provide valuable information to a fault monitoring system based on PCA. The PCA algorithm was utilized in this research to describe the relationships among the process variables. The fault directions for various fault scenarios were obtained using the SVD technique on the prediction errors, and fault isolation was then accomplished using new projections on these existing fault directions. The fault isolation index provides a convenient means of isolating the faults as it is close to unity on a particular direction for the corresponding fault whenever it occurs and is smaller for the others. The developed optimum sensor placement design formulations and the PCA-based fault diagnostic methods were demonstrated with application to the coupled IRIS-MSF nuclear desalination plant. When the optimal sensor sets were used, both sensor faults and process faults were correctly detected and isolated based on T^2 and Q statistics, as well as fault isolation index. Fault diagnostic results were also examined using the reduced sensor sets for the same fault scenarios, and the results indicated that the reduced sensor sets were not enough for the purpose of fault isolation. Another important discovery through the FDI case studies was that the PCA fault diagnostic approach could be extended to detect and isolate simultaneous dual faults using the optimal sensor sets identified for the single-fault cases, in which case both single-fault and dual-fault directions were used as fault signatures for dual-fault isolation. The results demonstrated the effectiveness of the developed methods for fault detection and isolation in complex systems with optimized sensor network.

6. The results of this R&D also showed that it would be sufficient to use the optimal sensor set to build effective PCA models for FDI. It is not necessary to use all the measurements in the database to develop a PCA representation, but those that are given through optimal sensor selection algorithms. This eliminates the arbitrary decision making in the selection of measurements needed for PCA modeling.

9.2 Suggestions for Future Research

The area of FDI holds many opportunities for continuing research beyond the scope of this dissertation. Although demonstrated through the simulation data generated by a developed SIMULINK model of the IRIS-MSF nuclear desalination plant, the developed FDI algorithms need to be implemented and tested on a laboratory system and possibly using real nuclear desalination plant data. This would indicate areas where further development is necessary to establish a more efficient fault monitoring system. As part of future research, it is suggested that the fault diagnostic procedure be expanded to more fault scenarios and attempts be made to automate the procedure.

Sensor placement design based on directed graph was presented to determine the minimum set of sensors for fault detection and isolation, and was applied to the IRIS-MSF nuclear desalination system. In this dissertation, the DG models were developed manually from engineering models. To facilitate computer aided design, this process should be automated by developing a generic software package that allows an automatic transformation of engineering models into DGs and the determination of minimum requirements of sensor placement from a fault diagnosis perspective.

Future research also should consist of the use of other optimization criteria for sensor placement design, such as process controllability constraint, residual precision, data reconciliation, and gross error robustness, as well as the treatment of the sensor placement design as a multi-objective optimization problem. Development of an integrated sensor network design strategy will make the seamless incorporation of various optimization criteria possible, so that the resulting sensor placement is optimal in a much broader sense, as opposed to the sole consideration of fault diagnostics.

List of References

1. Abdel-Jabbar, N.M., H.M. Qiblawey, F.S. Mjalli, and H. Ettouney (2007), "*Simulation of large capacity MSF brine circulation plants*," Desalination, vol. 204, pp. 501-514, 2007.
2. Ahn, S.J., C.J. Lee, Y. Jung, C. Han, E.S. Yoon, and G. Lee (2008), "*Fault diagnosis of the multi-stage flash desalination process based on signed digraph and dynamic partial least square*," Desalination, vol. 228, pp. 68-83, 2008.
3. Aickelin, U. (2002), "*An indirect genetic algorithm for set-covering problem*," Journal of Operational Research Society, vol. 53, pp. 1118-1126, 2002.
4. Alatiqi, I., H. Ettouney, and H. El-Dessouky (1999), "*Process control in water desalination industry: an overview*," Desalination, vol. 126, pp. 15-32, 1999.
5. Ali, E., A. Ajbar, and K. Alhumaizi (1997), "*Robust control of industrial multi-stage flash desalination plants*," Desalination, vol. 114, pp. 289-302, 1997.
6. Ali, E., K. Alhumaizi, and A. Ajbar (1999), "*Model reduction and robust control of multi-stage flash (MSF) desalination plants*," Desalination, vol. 121, pp. 65-85, 1999.
7. Ali, E. (2002a), "*Understanding the operation of industrial MSF plants Part I: Stability and steady-state analysis*," Desalination, vol. 143, pp. 53-72, 2002.
8. Ali, E. (2002b), "*Understanding the operation of industrial MSF plants Part II: Optimization and dynamic analysis*," Desalination, vol. 143, pp. 73-91, 2002.
9. Ali, Y., and S. Narasimhan (1993), "*Sensor Network Design for Maximizing Reliability of Linear Processes*," American Institute of Chemical Engineering Journal, vol. 39, no. 5, pp. 820-828, 1993.
10. Ali, Y., and S. Narasimhan (1995), "*Redundant Sensor Network Design for Linear Processes*," American Institute of Chemical Engineering Journal, vol. 41, no. 10, pp. 2237-2249, 1995.
11. Ali, Y., and S. Narasimhan (1996), "*Sensor Network Design for Maximizing Reliability of*

- Bilinear Processes*,” American Institute of Chemical Engineering Journal, vol. 42, no. 9, pp. 2563–2575, 1996.
12. Aly, N.H., and M.A. Marwan (1995), “*Dynamic behavior of MSF desalination plants*,” Desalination, vol. 101, pp. 287-293, 1995.
 13. Aly, N.H., and M.A. Marwan (1997), “*Dynamic response of multi-effect evaporators*,” Desalination, vol. 114, pp. 189-196, 1997.
 14. Al-Shehabi, A.G., and B. Newman (2001), “*Aeroelastic vehicle optimal sensor placement for feedback control applications using mixed gain-phase stability*,” Proceedings of the American Control Conference, vol. 3, pp. 1848-1852, June 2001.
 15. Asratian, A.S., T.M.J. Denley, and R. Hagkvist (1998), *Bipartite Graphs and their Applications*, Cambridge University Press, UK, 1998.
 16. Azam, M., K. Pattipati, and A. Patterson-Hine (2004), “*Optimal Sensor Allocation for Fault Detection and Isolation*,” IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1309–1314, 2004.
 17. Bagajewicz, M. (1997), “*Design and Retrofit of Sensor Networks in Process Plants*,” American Institute of Chemical Engineering Journal, vol. 43, no. 9, pp. 2300–2306, 1997.
 18. Bagajewicz, M., and M. Sanchez (1999), “*Design and Upgrade of Nonredundant and Redundant Linear Sensor Networks*,” American Institute of Chemical Engineering Journal, vol. 45, no. 9, pp. 1927–1938, 1999.
 19. Bagajewicz, M., and M. Sanchez (2000), “*Cost-optimal Design of Reliable Sensor Networks*,” Computers and Chemical Engineering, vol. 23, pp. 1757–1762, 2000.
 20. Bagajewicz, M. (2001), *Process Plant Instrumentation: Design and Upgrade*, Technomic Publishing Company, Inc., 2001.
 21. Bagajewicz, M. (2002a), “*A Review of Techniques for Instrumentation Design and*

- Upgrade in Process Plants*,” The Canadian Journal of Chemical Engineering, vol. 80, no. 1, pp. 3-16, 2002.
22. Bagajewicz, M., and E. Cabrera (2002b), “*New MILP Formulation for Instrumentation Network Design and Upgrade*,” American Institute of Chemical Engineering Journal, vol. 48, no. 10, pp. 2271–2282, 2002.
 23. Bagajewicz, M., A. Fuxman, and A. Uribe (2004), “*Instrumentation Network Design and Upgrade for Process Monitoring and Fault Detection*,” American Institute of Chemical Engineering Journal, vol. 50, no. 8, pp. 1870–1880, 2004.
 24. Beasley, J.E. (1987), “*An algorithm for set-covering problem*,” European Journal of Operational Research, vol. 31, pp 85-93, 1987.
 25. Beasley, J.E. (1990), “*A Lagrangian heuristic for set-covering problems*,” Naval Research Logistics, vol. 37, pp. 151-164, 1990.
 26. Beasley, J.E., and P.C. Chu (1996), “*A genetic algorithm for the set-covering problem*,” European Journal of Operational Research, vol. 94, pp. 392-404, 1996.
 27. Bhushan, M., and R. Rengaswamy (2000a), “*Design of Sensor Network Based on the Signed Directed Graph of the Process for Efficient Fault Diagnosis*,” Industrial and Engineering Chemistry Research, vol. 39, pp. 999–1019, 2000.
 28. Bhushan, M., and R. Rengaswamy (2000b), “*Design of Sensor Location Based on Various Fault Diagnostic Observability and Reliability Criteria*,” Computers and Chemical Engineering, vol. 24, pp. 735–741, 2000.
 29. Bhushan, M., and R. Rengaswamy (2002a), “*Comprehensive Design of a Sensor Network for Chemical Plants Based on Various Diagnosability and Reliability Criteria-1. Framework*,” Industrial and Engineering Chemistry Research, vol. 41, pp. 1826–1839, 2002.
 30. Bhushan, M., and R. Rengaswamy (2002b), “*Comprehensive Design of a Sensor Network*

- for Chemical Plants Based on Various Diagnosability and Reliability Criteria-2. Applications,” Industrial and Engineering Chemistry Research, vol. 41, pp. 1840–1860, 2002.*
31. Bhushan, M., S. Narasimhan, and R. Rengaswamy (2008), “*Robust Sensor Network Design for Fault Diagnosis,*” *Computers and Chemical Engineering*, vol. 32, pp. 1067-1084, 2008.
 32. Bulsari, A.B. (1995), *Neural networks for chemical engineers*, Elsevier Science, Amsterdam, 1995.
 33. Carelli, M.D., L. Conway, L. Oriani, B. Petrovic, N.E. Todreas (2003), “*The Design and Safety Features of the IRIS Reactor,*” ICONE 11, Tokyo, Japan, April 20-23, 2003.
 34. Carelli, M.D., and B. Petrovic (2006), “*IRIS - progressing toward a worldwide deployment,*” *Proceedings of the International Conference on Advances in Nuclear Power Plants (ICAPP’06)*, Reno, NV, 2006.
 35. Carpenter, G.A., and S. Grossberg (1988), “*The art of adaptive pattern recognition by a self-organizing neural network,*” *Computer*, vol. 21, no. 3, pp. 77-88, 1988.
 36. Chen, B.H., X.Z. Wang, S.H. Yang, and C. Mcgreavy (1999), “*Application of wavelets and neural networks to diagnostic system development: I. Feature extraction,*” *Computers and Chemical Engineering*, vol. 23, no. 7, pp. 899-906, 1999.
 37. Chen, S.Y., and Y.F. Li (2002), “*A method of Automatic Sensor Placement for Robot Vision in Inspection Tasks,*” *Proceedings of the IEEE International Conference on Robotics & Automation*, vol. 3, pp. 2545-2550, 2002.
 38. Chmielewski, D., T. Palmer, and V. Manousiouthakis (2002), “*On the Theory of Optimal Sensor Placement,*” *American Institute of Chemical Engineering Journal*, vol. 48, no. 5, pp. 1001–1012, 2002.
 39. Chow, E.Y., and A.S. Willsky (1984), “*Analytical redundancy and the design of robust*

- detection system*,” IEEE Transactions on Automatic Control, vol. 29, pp. 603-614, 1984.
40. Cybenko, G. (1989), “*Approximation by superpositions of a sigmoidal function*,” Mathematics of Control, Signals, and Systems, vol. 2, no. 4, pp. 303-314, 1989.
 41. Dardour, S., S. Nisan, and F. Charbit (2005), “*Development of a computer-package for MED plant dynamics*,” Desalination, vol. 182, pp. 229-237, 2005.
 42. Delene, J.G., and S.J. Ball (1971), *A Digital Computer Code for Simulating Large Multistage Flash Evaporator Desalting Plant Dynamics*, Report ORNL-TM-2933, ORNL, Oak Ridge, TN, September 1971.
 43. Dochain, D., N. Tali-Maanar, and J.P. Babary (1997), “*On Modeling, Monitoring and Control of Fixed Bed Bioreactors*,” Computers and Chemical Engineering, vol. 21, pp. 1255–1266, 1997.
 44. Duda, R.O., and P.E. Hart (1973), *Pattern classification and scene classification*, Wiley Interscience, New York, 1973.
 45. Dunia, R., and S.J. Qin (1998), “*Subspace approach to multidimensional fault identification and reconstruction*,” American Institute of Chemical Engineering Journal, vol. 44, pp. 1813-1831, 1998.
 46. Eberhart, R.C., and J. Kennedy (1995), “*A new optimization using particle swarm theory*,” Proceedings of the 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39-43, 1995.
 47. El-Dessouky, H.T., H.M. Ettouney, and Y. Al-Roumi (1999), “*Multi-stage flash desalination: present and future outlook*,” Chemical Engineering Journal, vol. 73, no. 2, pp. 173-190, 1999.
 48. Erbay, A.S., and B.R. Upadhyaya (1997), “*A Personal Computer-Based On-Line Signal Validation System for Nuclear Power Plants*,” Nuclear Technology, vol. 119, pp. 63-75, 1997.

49. Fan, J.Y., M. Nikolaou, and R.E. White (1993), "*An approach to fault diagnosis of chemical processes via neural networks*," American Institute of Chemical Engineering Journal, vol. 39, no. 1, pp. 82-88, 1993.
50. Farrell, A.E., and S.D. Roat (1994), "*Framework for enhancing fault diagnosis capabilities of artificial neural networks*," Computers and Chemical Engineering, vol. 18, no. 7, pp. 613-635, 1994.
51. Farlow, S.J. (1984), *Self-Organizing Methods in Modeling: GMDH-Type Algorithms*, Marcel-Dekker, New York, 1984.
52. Fathi, Z., W.F. Ramirez, and J. Korbiez (1993), "*Analytical and knowledge-based redundancy for fault diagnosis in process plants*," American Institute of Chemical Engineering Journal, vol. 39, pp. 42-56, 1993.
53. Faulds, A.L., and B.B. King (2000), "*Sensor location in feedback control of partial differential equation systems*," Proceedings of the IEEE International Conference on Control Applications, vol. 1, pp. 536-541, Sept. 2000.
54. Flury, B. (1989), *Common Principal Components and Related Multivariate Models*, John Wiley, New York, 1989.
55. Furuya, H., and R.T. Haftka (1996), "*Combining Genetic and Deterministic Algorithms for Locating Actuators on Space Structures*," Journal of Spacecraft and Rockets, vol. 33, no. 3, pp. 422-427, 1996.
56. Gambier, A., and E. Badreddin (2002), "*Application of hybrid modeling and control techniques to desalination plants*," Desalination, vol. 152, pp. 175-184, 2002.
57. Gambier, A., and E. Badreddin (2005), "*Dynamic Modeling of a Single-stage MSF Plant for Advanced Control Purposes*," Proceedings of the IEEE Conference on Control Applications, Toronto, Canada, August 2005.
58. Giraud, C., and B. Jouvencel (1995), "*Sensor Selection: a Geometrical Approach*,"

- Proceedings of the IEEE/RSJ International Conference, vol. 2, pp. 555-560, 1995.
59. Glover, F., and M. Laguna (1997), *Tabu Search*, Kluwer Academic Publishers, Boston, 1997.
 60. Gluek, A.R., and R.W. Bradshaw (1970), “A mathematical model for a MSF distillation plant,” Proceedings of the 3rd International Symposium on Fresh Water from the Sea, vol. 1, pp. 95-108, 1970.
 61. Griffith, W.L., and R.M. Keller (1965), *Saline: A Fortran Computer Program for the Process Design of Saline Water Conversion Plants using the Multi-Stage, Flash-Evaporation Process*, Report ORNL-TM-1299, ORNL, Oak Ridge, TN, November 1965.
 62. Hac, A., and L. Liu (1993), “Sensor and Actuator Location in Motion Control of Flexible Structures,” *Journal of Sound and Vibration*, vol. 167, no. 2, pp. 239–261, 1993.
 63. Haddadi, S. (1997), “Simple Lagrangian heuristic for the set-covering problem,” *European Journal of Operational Research*, vol. 97, pp 200-204, 1997.
 64. Hastie, T., R. Tibshirani, and J. Friedman (2001), *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, New York, 2001.
 65. Helal, A.M., M.S. Medani, and M.A. Soliman (1986), “A Tridiagonal Matrix Model for Multistage Flash Desalination Plants,” *Computers and Chemical Engineering*, vol. 10, pp. 327-342, 1986.
 66. Hines, J.W., D.J. Wrest, and R.E. Uhrig (1997), “Signal Validation using Adaptive Neural Fuzzy Inference Systems,” *Nuclear Technology*, vol. 119, no. 2, pp. 181-193, 1997.
 67. Hines, J.W., and D. Garvey (2007), “Process and equipment monitoring methodologies applied to sensor calibration monitoring,” *Quality and Reliability Engineering International*, vol. 23, no. 1, pp. 123-135, 2007.

68. Hines, J.W. et al. (2008), *Technical Review of On-Line Monitoring Techniques for Performance Assessment*, vol. 1, 2 & 3, NUREG/CR-6895, May 2008.
69. Hines, J.W., B.R. Upadhyaya et al. (2008), *Advanced Instrumentation and Control Methods for Small and Medium Reactors with IRIS Demonstration*, Annual Report prepared for the U.S. Department of Energy, DE-FG07-07ID14895/UTNE/2008-4, October 2008.
70. Hines, J.W., B.R. Upadhyaya et al. (2009), *Advanced Instrumentation and Control Methods for Small and Medium Reactors with IRIS Demonstration*, Annual Report prepared for the U.S. Department of Energy, DE-FG07-07ID14895/UTNE/2009-4, October 2009.
71. Hines, J.W., B.R. Upadhyaya et al. (2010), *Advanced Instrumentation and Control Methods for Small and Medium Reactors with IRIS Demonstration*, Annual Report prepared for the U.S. Department of Energy, DE-FG07-07ID14895/UTNE/2010-4, October 2010.
72. Holbert, K.E., and B.R. Upadhyaya (1990), "An Integrated Signal Validation System for Nuclear Power Plants," *Nuclear Technology*, vol. 92, no. 3, pp. 411-427, 1990.
73. Holbert, K.E., and B.R. Upadhyaya (1994), "Empirical Process Modeling Technique for Signal Validation," *Annals of Nuclear Energy*, vol. 21, no. 7, pp. 387-403, 1994.
74. Humberstone, M.J. (2010), *An Adaptive Nonparametric Modeling Technique for Expanded Condition Monitoring of Processes*, Ph.D. Dissertation, University of Tennessee, Knoxville, May 2010.
75. Husain, A., A. Hassan, D.M.K. Al-Gobaisi, A. Al-Radif, A. Woldai, and C. Sommariva (1993), "Modelling, simulation, optimization and control of multistage flashing (MSF) desalination plants Part I: modelling and simulation," *Desalination*, vol. 92, pp. 21-41, 1993.

76. Husain, A., A. Woldai, A. Al-Radif, A. Kesou, R. Borsani, H. Sultan, and P.B. Deshpandey (1994), "*Modelling and simulation of a multistage flash (MSF) desalination plant*," Desalination, vol. 97, pp. 555-586, 1994.
77. International Atomic Energy Agency (2002), *Design Concepts of Nuclear Desalination Plants*, TECDOC-1326, IAEA, Austria, November 2002.
78. International Atomic Energy Agency (2005), *Optimization of the Coupling of Nuclear Reactors and Desalination Systems*, TECDOC-1444, IAEA, Austria, June 2005.
79. International Atomic Energy Agency (2007), *Status of Nuclear Desalination in IAEA Member States*, TECDOC-1524, IAEA, Austria, January 2007.
80. Ingersoll, D.T., J.L. Binder, D. Conti, and M.E. Ricotti (2004), "*Nuclear Desalination Options for the international Reactor Innovative and Secure (IRIS) Design*," Proceedings of the 5th International Conference on Nuclear Option in Countries with Small and Medium Electricity Grids, Dubrovnik, Croatia, May 2004.
81. Ingersoll, D.T. (2009), "*Deliberately small reactors and the second nuclear era*," Progress in Nuclear Energy, vol. 51, pp. 589-603, 2009.
82. International Water Management Institute (2000), *Vision of Water for Food and Rural Development*, World Water Forum, Hague, March 2000.
83. Jackson, J.E., and G.S. Mudholkar (1979), "*Control procedures for residuals associated with principal component analysis*," Technometrics, vol. 21, pp. 341-349, 1979.
84. Jackson, J.E. (1991), *A user guide to principal components*, John Wiley & Sons, New York, 1991.
85. James, M. (1985), *Classification Algorithms*, John Wiley, New York, 1985.
86. Johnson, D.S., C.R. Aragon, L.A. McGeoch, and C. Schevon (1989), "*Optimization by Simulated Annealing: An Experimental Evaluation; Part I, Graph Partitioning*,"

- Operations Research, vol. 37, pp. 865-893, 1989.
87. Jolliffe, I.T. (2002), *Principal Component Analysis, 2nd edition, Springer Series in Statistics*, Springer, New York, 2002.
 88. Kadu, S.C., M. Bhushan, and R. Gudi (2008), "Optimal sensor network design for multirate systems," *Journal of Process Control*, vol. 18, pp. 594-609, 2008.
 89. Kaistha, N., and B.R. Upadhyaya (2001), "Incipient fault detection and isolation of field devices in nuclear power systems using principal component analysis," *Nuclear Technology*, vol. 136, pp. 221-230, 2001.
 90. Kalman, R.E. (1960), "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 1, pp. 35-45, 1960.
 91. Kavaklioglu, K., and B.R. Upadhyaya (1994), "Monitoring feedwater flow rate and component thermal performance of pressurized water reactors by means of artificial neural networks," *Nuclear Technology*, vol. 107, pp.112–123, 1994.
 92. Khan, A.H. (1986), *Desalination Processes and Multistage Flash Distillation Practice*, Elsevier Publishers, New York, 1986.
 93. Kohonen, T. (1984), *Self-organization and associative memory*, Springer, New York, 1984.
 94. Kotecha, P.R., M. Bhushan, R.D. Gudi, and M.K. Keshari (2008a), "A Duality Based Framework for Integrating Reliability and Precision for Sensor Network Design," *Journal of Process Control*, vol. 18, pp. 189-201, 2008.
 95. Kotecha, P.R., M. Bhushan, and R.D. Gudi (2008b), "Design of robust, reliable sensor networks using constraint programming," *Computers and Chemical Engineering*, vol. 32, pp. 2030-2049, 2008.
 96. Kramer, M.A., and B.L. Palowitch (1987), "A rule-based approach to fault diagnosis using

- the signed directed graph*,” American Institute of Chemical Engineering Journal, vol. 33, pp. 1067-1078, 1987.
97. Kretsovalis, A., and R. Mah (1987), “*Observability and Redundancy Classification in Multicomponent Process Networks*,” American Institute of Chemical Engineering Journal, vol. 33, no. 1, pp. 70–82, 1987.
 98. Krysander, M., and E. Frisk (2008), “*Sensor placement for fault diagnosis*,” IEEE Transaction on Systems, Man, and Cybernetics-Part A, vol. 38, no. 6, pp. 1398-1410, 2008.
 99. Lambert, H.E. (1977), “*Fault Trees for Locating Sensors in Process Systems*,” Chemical Engineering Progress, vol. 73, no. 8, pp. 81-85, 1977.
 100. Lim, K.B. (1997), “*A Disturbance Rejection Approach to Actuator and Sensor Placement*,” Journal of Guidance, Control, and Dynamics, vol. 20, no. 1, pp. 202-204, 1997.
 101. Lior, N., R. Chung, and O. Miyatake (2002), “*Correlations (updated) for predicting the flow through MSF plant interstage orifices*,” Desalination, vol. 151, pp. 209-216, 2002.
 102. Luong, M., D. Maquin, C. Huynh, and J. Ragot (1994), “*Observability, Redundancy, Reliability and Integrated Design of Measurement System*,” Proceedings of the 2nd IFAC Symposium on Intelligent Components and Instruments for Control Applications, SICICA'94, 1994.
 103. Madron, F., and V. Veverka (1992), “*Optimal Selection of Measuring Point in Complex Plants by Linear Models*,” American Institute of Chemical Engineering Journal, vol. 38, no. 2, pp. 227–236, 1992.
 104. Madron, F., J. Papuga, and J. Pliska (2006), “*Thermal Performance Monitoring and Assessment in Dukovany Nuclear Power Plant*,” Proceedings of the 5th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human Machine Interface

- Technologies (NPIC&HMIT 2006), Albuquerque, NM, 2006.
105. Maniar, V.M., and P.B. Deshpande (1996), “*Advanced controls for multi-stage flash (MSF) desalination plant optimization*,” Journal of Process Control, vol. 6, no. 1, pp. 49-66, 1996.
 106. Mattison, B., M. Naghedolfeizi, and B.R. Upadhyaya (2007), “*Development of a Graphical User Interface for a Dynamic Model of a Pressurized Water Reactor Plant*,” M.S. Project Report, University of Tennessee, Knoxville, 2007.
 107. Maul, W.A., G. Kopasakis, L.M. Santi, T.S. Sowers, and A. Chicatelli (2007), *Sensor Selection and Optimization for Health Assessment of Aerospace Systems*, NASA/TM-2007-214822, 2007.
 108. Maurya, M.R., R. Rengaswamy, and V. Venkatasubramanian (2006), “*A Signed Directed Graph-based Systematic Framework for Steady-State Malfunction Diagnosis inside Control Loops*,” Chemical Engineering Science, vol. 61, pp. 1790-1810, 2006.
 109. Maybeck, P.S. (1979), *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, New York, 1979.
 110. Mazzotti, M., M. Rosso, A. Beltramini, and M. Morbidelli (2000), “*Dynamic modeling of multistage flash desalination plants*,” Desalination, vol. 127, pp. 207-218, 2000.
 111. Mazzour, E.H., D. Hodouin, and S. Makni (2003), “*Optimal Sensor Implementation in Metallurgical Plants-An Application to a Generic Mineral Separation Plant*,” International Journal of Mineral Processing, vol. 69, pp. 185-203, 2003.
 112. Megahed, M.M. (2001), “*Nuclear desalination: history and prospects*,” Desalination, vol. 135, pp. 169-185, 2001.
 113. Monfroglio, A. (1998), “*Hybrid heuristic algorithm for set-covering*,” Computers Operational Research, vol. 25, pp. 441-445, 1998.
 114. Muller, P., and H. Weber (1972), “*Analysis and Optimization of Certain Qualities of*

- Controllability and Observability for Linear Dynamical Systems*,” Automatica, vol. 8, pp. 237–246, 1972.
115. Mushini, R., and D. Simon (2005), “*On Optimization of Sensor Selection for Aircraft Gas Turbine Engines*,” Proceedings of the 18th International Conference on Systems Engineering, ISCEng’05, IEEE, pp. 9–14, 2005.
 116. Musulin, E., C. Benqlilou, M. Bagajewicz, and L. Puigjaner (2005), “*Instrumentation Design Based on Optimal Kalman Filtering*,” Journal of Process Control, vol. 15, no. 6, pp. 629–638, 2005.
 117. Naghedolfeizi, M. (1990), *Dynamic Modeling of a Pressurized Water Reactor Plant for Diagnostics and Control*, M.S. Thesis, University of Tennessee, Knoxville, 1990.
 118. Naimimohasses, R., D.M. Barnett, D.A. Green, and P.R. Smith (1995), “*Sensor optimization using neural network sensitivity measures*,” Measurement Science and Technology, vol. 6, pp.1291-1300, 1995.
 119. Narasimhan, S., P. Mosterman, and G. Biswas (1998), “*A Systematic Analysis of Measurement Selection Algorithms for Fault Isolation in Dynamic Systems*,” Proceedings of the 9th International Workshop on Principles of Diagnosis, pp. 94–101, 1998.
 120. Nguyen, D.Q., and M.J. Bagajewicz (2008), “*Design of Nonlinear Sensor Networks for Process Plants*,” Industrial and Engineering Chemistry Research, vol. 47, no. 15, pp. 5529-5542, 2008.
 121. Padula, S.L., and R.K. Kincaid (1999), *Optimization strategies for sensor and actuator placement*, NASA/TM-1999-209126, April 1999.
 122. Papadopoulos, M., and E. Garcia (1998), “*Sensor Placement Methodologies for Dynamic Testing*,” AIAA Journal, vol. 36, no. 2, pp. 256–263, 1998.
 123. Parker, R.G., and R. L. Rardin (1988), *Discrete Optimization*, Academic Press, San Diego, 1988.

124. Ponslet, E., R.T. Haftka, and H.H. Cudney (1993), "*Optimal Placement of Tuning Masses on Truss Structures by Genetic Algorithm*," Collection of Technical Papers - AIAA/ASME Structures, Dynamics and Materials Conference, Part 4, pp. 2448-2457, 1993.
125. Raghuraj, R., M. Bhushan, and R. Rengaswamy (1999), "*Locating sensors in complex chemical plants based on fault diagnostic observability criteria*," American Institute of Chemical Engineering Journal, vol. 45, no. 2, pp. 310-322, 1999.
126. Rimawi, M.A., H.M. Ettouney, and G.S. Aly (1989), "*Transient Model of Multistage Flash Desalination*," Desalination, vol. 74, pp. 327-338, 1989.
127. Rosso, M., A. Beltramini, M. Mazzotti, and M. Morbidelli (1996), "*Modeling multistage flash desalination plants*," Desalination, vol. 108, pp. 365-374, 1996.
128. Santi, L.M, T.S. Sowers, and R.B. Aguilar (2005), *Optimal Sensor Selection for Health Monitoring Systems*, NASA/TM-2005-213955, 2005.
129. Sen, S., S. Narasimhan, and K. Deb (1998), "*Sensor Network Design of Linear Processes Using Genetic Algorithms*," Computers and Chemical Engineering, vol. 22, no. 3, pp. 385-390, 1998.
130. Shankar, G. (1977), "*Simulation of a Nuclear Turbine*," Journal of Nuclear Engineering and Design, vol. 44, pp. 269-277, 1977.
131. Shiozaki, J., B. Shibata, H. Matsuyama, and E. O'Shima (1989), "*Fault Diagnosis of Chemical Processes Utilizing Signed Directed Graphs-Improvement by Using Temporal Information*," IEEE Transactions on Industrial Electronics, vol. 36, no. 4, pp. 469-474, 1989.
132. Sorenson, H.W. (1970), "*Least-Squares estimation: from Gauss to Kalman*," IEEE Spectrum, vol. 7, no. 7, pp. 63-68, 1970.
133. Spanache, S., T. Escobet, and L. Travé-Massuyès (2004), "*Sensor Placement Optimisation Using Genetic Algorithms*," Proceedings of the 15th International Workshop on Principles

- of Diagnosis, pp. 179–183, 2004.
134. Tanvir, M.S., and I.M. Mujtaba (2008), “*Optimisation of design and operation of MSF desalination process using MINLP technique in gPROMS*,” *Desalination*, vol. 222, pp. 419-430, 2008.
 135. Tarifa, E.E., and N.J. Scenna (1997), “*Fault diagnosis, direct graphs, and fuzzy logic*,” *Computers and Chemical Engineering*, vol. 21, pp. 649-654, 1997.
 136. Tarifa, E.E., and N.J. Scenna (1998a), “*A methodology for fault diagnosis in large chemical processes and an application to a multistage flash desalination process: Part I*,” *Reliability Engineering and System Safety*, vol. 60, pp. 29-40, 1998.
 137. Tarifa, E.E., and N.J. Scenna (1998b), “*A methodology for fault diagnosis in large chemical processes and an application to a multistage flash desalination process: Part II*,” *Reliability Engineering and System Safety*, vol. 60, pp. 41-51, 1998.
 138. Tarifa, E.E., and N.J. Scenna (2001), “*A dynamic simulator for MSF plants*,” *Desalination*, vol. 138, pp. 349-364, 2001.
 139. Tarifa, E.E., and N.J. Scenna (2002), “*Fault diagnosis for a MSF using a SDG and fuzzy logic*,” *Desalination*, vol. 152, pp. 207-214, 2002.
 140. Tarifa, E.E., and N.J. Scenna (2004), “*Fault diagnosis for MSF dynamic states using a SDG and fuzzy logic*,” *Desalination*, vol. 166, pp. 93-101, 2004.
 141. Thanomsat, C., C.W. Taft, and A.M. Annaswamy (1998), “*Level Control in Feedwater Heater Systems Using Nonlinear Strategies*,” *ISA Transactions*, vol. 37, pp. 299-312, 1998.
 142. Thomas, P.J., S. Bhattacharyya, A. Patra, and G.P. Rao (1998), “*Steady state and dynamic simulation of multi-stage flash desalination plants: A case study*,” *Computers and Chemical Engineering*, vol. 22, pp. 1515-1529, 1998.

143. Tsai, C.S., and C.T. Chang (1995), "*Dynamic process diagnosis via integrated neural networks*," Computers and Chemical Engineering, vol. 19, pp. 747-752, 1995.
144. Tylee, J.L. (1983), "*On-Line Failure Detection in Nuclear Power Plant Instrumentation*," IEEE Transactions on Automatic Control, vol. AC-28, no. 3, pp. 406-415, 1983.
145. Udawadia, F.E. (1994), "*Methodology for Optimum Sensor Locations for Parameter Identification in Dynamic Systems*," Journal of Engineering Mechanics, vol. 120, no. 2, pp. 368-390, 1994.
146. Upadhyaya, B.R. et al. (1989), *Development and Testing of an Integrated Signal Validation System for Nuclear Power Plants*, Final Report prepared for the U.S. Department of Energy, DOE/NE/37959-36, September 1989.
147. Upadhyaya, B.R., and E. Eryurek (1992), "*Application of Neural Networks for Sensor Validation and Plant Monitoring*," Nuclear Technology, vol. 97, no. 2, pp. 170-176, 1992.
148. Vaclavek, V., and M. Loucka (1976), "*Selection of measurements necessary to achieve multi-component mass balances in chemical plant*," Chemical Engineering Science, vol. 31, no. 12, pp. 1199-1205, 1976.
149. Valle, S., W. Li, and S.J. Qin (1999), "*Selection of the number of principal components: the variance of the reconstruction error criterion with a comparison to other methods*," Industrial and Engineering Chemistry Research, vol. 38, pp. 4389-4401, 1999.
150. Van den Berg, F., H. Hoefsloot, H. Boelens, and A. Smilde (2000), "*Selection of Optimal Sensor Position in a Tubular Reactor Using Robust Degree of Observability Criteria*," Chemical Engineering Science, vol. 55, pp. 827-837, 2000.
151. Vasko, J.F., and F.E. Wolf (2001), "*A heuristic concentration approach for weighted set-covering problems*," Locator: ePublication of Location Analysis, vol. 2, no. 1, pp. 1-14, 2001.
152. Venkatasubramanian, V., and T.J. McAvoy (1992), "*Special issue on neural network*

- applications in chemical engineering,” Computers and Chemical Engineering*, vol. 16, no. 4, 1992.
153. Wang, H., Z. Song, and P. Li (2000), “*Improved PCA with Optimized Sensor Locations for Process Monitoring and Fault Diagnosis*,” Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, December 2000.
 154. Wang, H., Z. Song, and H. Wang (2002a), “*Statistical Process Monitoring Using Improved PCA with Optimized Sensor Locations*,” *Journal of Process Control*, vol. 12, pp. 735-744, 2002.
 155. Wang, H., Z. Song, and H. Wang (2002b), “*Fault detection behavior analysis of PCA-based process monitoring approach*,” *Journal of Chemical Industry and Engineering*, vol. 53, no. 3, pp. 297–301, 2002.
 156. Wang, H., Z. Song, and P. Li (2002c), “*Fault Detection Behavior and Performance Analysis of Principal Component Analysis Based Process Monitoring Methods*,” *Industrial and Engineering Chemistry Research*, vol. 41, pp. 2455-2464, 2002.
 157. Wang, X., G. Rong, and J. Li (2007), “*A New Approach to Design Reliable General Sensor Network on the Basis of Graph Theory*,” *Industrial and Engineering Chemistry Research*, vol. 46, no. 8, pp. 2520-2525, 2007.
 158. Wang, X.Z., B.H. Chen, S.H. Yang, and C. Mcgreavy (1999), “*Application of wavelets and neural networks to diagnostic system development: 2. An integrated framework and its application*,” *Computers and Chemical Engineering*, vol. 23, no. 7, pp. 945-954, 1999.
 159. Watanabe, K., S. Hirota, L. Hou, and D.M. Himmelblau (1994), “*Diagnosis of Multiple Simultaneous Faults via Hierarchical Artificial Neural Networks*,” *American Institute of Chemical Engineering Journal*, vol. 40, no. 5, pp. 839-848, 1994.
 160. Whiteley, J.R., and J.F. Davis (1994), “*A similarity-based approach to interpretation of sensor data using adaptive resonance theory*,” *Computers and Chemical Engineering*, vol.

18, no. 7, pp. 637-661, 1994.

161. Xu, J., and B.R. Upadhyaya (2009), "*Multivariable Control Design for the IRIS Primary System Using the Model-Predictive Control Approach*," Proceedings of the 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human Machine Interface Technologies (NPIC&HMIT 2009), Knoxville, TN, April 5-9, 2009.
162. Yan, Y. (2004), "*Sensor Placement and Diagnosability Analysis at Design Stage*," National Research Council of Canada, NRC 47160, 2004.
163. Zhang, G. (2005), *Optimum Sensor Localization/Selection in a Diagnostic/Prognostic Architecture*, Ph.D. Dissertation, Georgia Institute of Technology, January 2005.
164. Zhang, G., and G. Vachtsevanos (2007), "*A Methodology for Optimum Sensor Localization/Selection in Fault Diagnosis*," Proceedings of the IEEE Aerospace Conference, 2007.
165. Zhang, Y. (2004), *Product quality modeling & control based on vision inspection with an application to baking processes*, Ph.D. Dissertation, Georgia Institute of Technology, December 2004.
166. Zhao, K. (2005a), *An Integrated Approach to Performance Monitoring and Fault Diagnosis of Nuclear Power Systems*, Ph.D. Dissertation, University of Tennessee, Knoxville, May 2005.
167. Zhao, K., and B.R. Upadhyaya (2005b), "*Adaptive fuzzy inference causal graph approach to fault detection and isolation of field devices in nuclear power plants*," Progress in Nuclear Energy, vol. 46, pp. 226-240, 2005.

Appendices

Appendix A: Thermo-Physical Properties

Pressure of saturated steam

$$\log_{10} \frac{P_c}{P} = \frac{X}{T_s} \left(\frac{a + bX + cX^3}{1 + dX} \right)$$

where

$$a = 3.2437814$$

$$b = 5.86826 \times 10^{-3}$$

$$c = 1.1702379 \times 10^{-8}$$

$$d = 2.1878462 \times 10^{-3}$$

P = pressure (atm)

P_c = critical pressure (218.167 atm)

T_c = critical temperature (647.27 K)

$$X = T_c - T_s$$

The above equation is specified for temperatures from 50 to 300 °F

Specific enthalpy of saturated steam

$$h_v = 1059.237 + 0.4833T - 2.558 \times 10^{-4} T^2$$

where

h_v = specific enthalpy of saturated steam (BTU/lb)

T = temperature (°F)

Specific heat capacity of pure water

$$Cp_w = 1.0011833 - 6.1666652 \times 10^{-5} T + 1.3999989 \times 10^{-7} T^2 + 1.3333336 \times 10^{-9} T^3$$

where

Cp_w = specific heat capacity of water (BTU/lb · °F)

T = boiling temperature of water (°F)

The above expression for the heat capacity of saturated water is adequate for computations within the temperature range of 50 ~ 300 °F .

Specific enthalpy of saturated water

$$h_w = -31.92 + 1.0011833T - 3.0833326 \times 10^{-5} T^2 + 4.6666663 \times 10^{-8} T^3 + 3.333334 \times 10^{-10} T^4$$

where

h_w = specific enthalpy of saturated water (BTU/lb)

T = boiling temperature of water (°F)

Latent heat of vaporization of water

$$\lambda_s = h_v - h_w$$

where

λ_s = latent heat of vaporization (BTU/lb)

h_v = specific enthalpy of saturated steam (BTU/lb)

h_w = specific enthalpy of saturated water (BTU/lb)

Specific heat capacity of brine

$$Cp_b = [1.0 - X_b(0.011311 - 0.00001146T_b)]Cp_w$$

where

Cp_b = specific heat capacity of brine (BTU/lb · °F)

Cp_w = specific heat capacity of water (BTU/lb · °F)

T_b = brine temperature (°F)

X_b = salt concentration (wt%)

Density of brine

The expression for the brine density given here is valid for the range of 0~26% concentration and 40~300 °F. Density of pure water is calculated from the given equation putting $X_b=0$.

$$\begin{aligned} \rho_b = & 62.707172 + 49.364088X_b - 0.43955304 \times 10^{-2}T_b - 0.032554667X_bT_b \\ & - 0.46076921 \times 10^{-4}T_b^2 + 0.63240299 \times 10^{-4}X_bT_b^2 \end{aligned}$$

where

ρ_b = density (lb/ft³)

T_b = brine temperature (°F)

X_b = salt concentration (mass fraction)

Boiling point elevation

$$\begin{aligned} BPE = & 1.8 \times [565.757/T_K - 9.81559 + 1.54739 \ln T_K - (337.178/T_K - 6.41981 + 0.922753 \ln T_K)X \\ & + (32.681/T_K - 0.55368 + 0.079022 \ln T_K)X^2] \times [X / (266919.6/T_K^2 - 379.669/T_K + 0.334169)] \end{aligned}$$

where

BPE = boiling point elevation (°F)

$T_K = (T_b + 460)/1.8$ (°K)

T_b = brine temperature (°F)

$X = (19.819X_b)/(1 - X_b)$

X_b =salt concentration (mass fraction)

Non-equilibrium allowance

$$NEA = 352 H_j^{1.1} \Delta T_{B_j}^{-0.25} (\omega_j \times 10^{-3})^{0.5} T_{S_j}^{-2.5}$$

where

NEA = non – equilibrium allowance (°F)

H_j = height of brine pool in stage j (in.)

$\Delta T_{B_j} = T_{B_{j-1}} - T_{B_j}$ (°F)

$\omega_j = \frac{W}{w_j}$ = chamber load lb/hr per ft width

w_j = width of stage j (ft)

W = total mass circulated in the system (recycle R + make-up F, lb/hr)

Temperature loss across the demister and condenser tubes

$$\Delta = \exp(1.885 - 0.02063 T_{p_j})$$

where

T_{p_j} = temperature of the distillate product (°F)

Overall heat transfer coefficient

The data submitted by the Office of Saline Water (OSW), based on the Point Loma Demonstration Plant, have been correlated to calculate the sum of the different resistances included in the classical equation for U, except the inside film resistance [Griffith and Keller, 1965]. The correlation ties those resistances with the saturation temperature prevailing in the flash chamber. The polynomial fit is expressed as follows:

$$z = 0.1024768 \times 10^{-2} - 0.7473939 \times 10^{-5} T_{p_j} + 0.999077 \times 10^{-7} T_{p_j}^2 \\ - 0.430046 \times 10^{-9} T_{p_j}^3 + 0.6206744 \times 10^{-12} T_{p_j}^4$$

where

T_{p_j} = temperature of the distillate product (°F), the above equation is specified for temperatures ranging from 100 to 250 °F .

$$y = \frac{(v_j \times ID_j)^{0.2}}{(160 + 1.92 T_{F_j}) v_j}$$

where

v_j = linear velocity of brine stream (ft/sec)

ID_j=tube inner diameter (in.)

T_{Fj}=temperature of the cooling brine at the exit of condenser (°F)

From the values of z and y, the overall heat transfer coefficient (BTU/hr · °F · ft²) is calculated by the equation

$$U = \frac{1}{z + y}$$

Inter-stage flow rate

$$Q = (Kwh\sqrt{\Delta P\rho/1000}/1.96\times 10^{-8})/0.45359$$

$$\Delta P = P_{i-1} - P_i + 1.0\times 10^{-5}\rho g(L - Ch)$$

$$C = 0.61 + 0.18x - 0.58x^2 + 0.7x^3$$

$$x = \frac{gh\rho/1000}{100(P_{i-1} - P) + gL\rho/1000}$$

where

Q=inter-stage flow rate (lb/hr)

w=orifice width (m)

h=orifice height (m)

L=liquid level in the upstream stage (m)

ρ=liquid density (kg/m³)

P=stage pressure (bar)

K=orifice discharge coefficient

C=orifice contraction coefficient

Appendix B: IRIS Nuclear System Mathematical Model

1. Reactor Core

Point Reactor Dynamic Model

$$\frac{d(P/P_0)}{dt} = \frac{\rho - \beta_t}{\Lambda} \frac{P}{P_0} + \lambda C$$

$$\frac{dC}{dt} = \frac{\beta}{\Lambda} \frac{P}{P_0} - \lambda C$$

Reactor Core Heart Transfer Model (Mann's Model)

$$\frac{dT_F}{dt} = \frac{F_r P_0}{(MC_p)_F} \frac{P}{P_0} + \frac{U_{FC} A_{FC}}{(MC_p)_F} (\theta_1 - T_F)$$

$$\frac{d\theta_1}{dt} = \frac{(1-F_r) P_0}{2(MC_p)_c} \frac{P}{P_0} + \frac{U_{FC} A_{FC}}{(MC_p)_C} (T_F - \theta_1) + \frac{W_C (T_{cold} - \theta_1)}{2M_C}$$

$$\frac{d\theta_2}{dt} = \frac{(1-F_r) P_0}{2(MC_p)_C} \frac{P}{P_0} + \frac{U_{FC} A_{FC}}{(MC_p)_C} (T_F - \theta_1) + \frac{W_C (\theta_1 - \theta_2)}{2M_C}$$

$$\frac{dT_{hot}}{dt} = \frac{W_C (\theta_2 - T_{hot})}{M_{up}}$$

Variable	Definition
A_{FC}	Effective heat transfer area between fuel and coolant
C	Precursor concentration
C_{pC}	Coolant heat capacity
C_{pF}	Fuel heat capacity
F_r	Fraction of the total power generated in fuel elements
U_{FC}	Average overall heat transfer coefficient
M_C	Coolant mass in the core
ρ_C	Coolant density
M_F	Fuel mass in the core
M_{up}	Coolant mass in the upriser
W	Coolant flow rate
P	Reactor Core power
T_F	Fuel temperature
T_{cold}	Cold part temperature
T_{hot}	Hot part temperature
α_C	Coolant coefficient of reactivity

α_F	Fuel coefficient of reactivity
β	Delayed neutron group fraction (one group)
λ	Average of six group decay constant
Λ	Neutron generation time
ρ	Total reactivity
ρ_{ex}	External reactivity
ρ_{fb}	Feedback reactivity

Note:

1 lbm = 0.4732 kg

1 BTU = 1.055E3 Joule

1 ft = 0.3048 m

1 ft² = 0.0929 m²

F = 9/5 * C + 32

2. Helical Coil Steam Generator

A dynamic process is generally modeled as a distributed parameter system characterized by a set of partial differential equations. It is usually rather complicated to solve such a time dependent system with spatial variations. For this reason, a lumped model is used to describe the HCSG dynamic behavior. Each lump has the same averaged properties, so the spatial dependence can be represented simply by the interaction between adjacent lumps. In addition to the assumptions implied in a lumped model, the other major assumptions are as follows:

- Only one pressure is used to characterize the superheated region.
- The superheated vapor satisfies ideal gas law modified by an expansion coefficient.
- The temperature of the second node in the subcooled region is equal to the saturated temperature.
- The pressure drop between superheated region and saturated region is constant during any perturbation
- The pressure drop between the saturated region and the subcooled region is constant during any perturbation
- The steam quality in the boiling region can be assumed as a linear function of the axial coordinate so the density in the boiling region can be approximated as a function of steam pressure.
- The steam generation rate assumes to be equal to the boiling rate.
- The heat transfer coefficient for the superheated region, the saturated region and the subcooled region is assumed constant.

2.1 Nodalization

Three regions, subcooled region, saturated region and superheated region, are used to characterize the significant difference of heat transfer and hydraulic behavior. In each region, two lumps with each volume are used to consider the axial temperature changes. Correspondingly, six metal nodes are needed to describe the heat transfer from the primary side

to the secondary side. For the two lumps of the secondary side in the saturated region, saturated temperature is maintained.

2.2 Primary Side Equations

The primary side temperature is given by:

$$\frac{dT_{pi}}{dt} = a_1 \frac{T_{pi-1} - T_{pi}}{L_i} - a_2 (T_{pi} - T_{wi})$$

where

$$L_i = \begin{cases} L_s & \text{for } i = 1, 2 \\ L_b & \text{for } i = 3, 4 \\ L_{sc} & \text{for } i = 5, 6 \end{cases}$$

$$a_1 = \frac{C_{pp} W_p}{(A_{xs} C_p \rho)_p / 2}$$

$$a_2 = \frac{h_{pw} P_{wp}}{(A_{xs} C_p \rho)_p / 2}$$

L_s = superheated length.

L_b = boiling length.

L_{sc} = subcooled length.

T = primary side temperature.

W_p = coolant flow rate.

C_p = specific heat.

ρ = density of the primary coolant.

A_{xs} = flow area.

h = heat transfer coefficient.

P_w = perimeter for heating.

In the above equations, subscript p and w refer to primary coolant and tube wall respectively.

2.3 Metal Equations

The metal temperature is given by:

$$\frac{d}{dt} T_{wi} = a_3 (T_{pi} - T_{wi}) - a_{4i} (T_{wi} - T_{si}) + (T_{wi} - T_{wi-1}) \dot{Z}_{i-1} / L_i + (T_{wi+1} - T_{wi}) \dot{Z}_i / L_i$$

for $i = 1, 2, 3, 4, 5, 6$

where

$$a_3 = \frac{h_{pw} P_{wp}}{(\rho_w A_w c_{Pw})}$$

$$a_{4i} = \begin{cases} \frac{h_{ws} P_{ws}}{(\rho_w A_w c_{pw})} & i = 1, 2 \\ \frac{h_{wb} P_{wb}}{(\rho_w A_w c_{pw})} & i = 3, 4 \\ \frac{h_{wb} P_{wb}}{(\rho_w A_w c_{pw})} & i = 5, 6 \end{cases}$$

$$\dot{Z}_0 = \dot{Z}_6 = 0$$

$$\dot{Z}_1 = \dot{L}_s / 2$$

$$\dot{Z}_2 = \dot{L}_s$$

$$\dot{Z}_3 = -(\dot{L}_{sc} + \dot{L}_b / 2)$$

$$\dot{Z}_4 = -\dot{L}_{sc}$$

$$\dot{Z}_5 = -\dot{L}_{sc} / 2$$

where

h_{ws}, h_{wb}, h_{wsc} = heat transfer coefficient on the tube side for superheated steam region, saturated water region, and subcooled water region respectively

P_{ws}, P_{wb}, P_{wsc} = heating circumference on the tube side for superheated steam region, saturated water region, and subcooled water region respectively

2.4 Equations for the Superheated Region

The mass balance of the steam in the superheated steam nodes, node 1 and node 2, are given by:

$$\dot{M}_{s1} = W_{21} - W_s$$

$$\dot{M}_{s2} = W_b - W_{21}$$

where

M_s = steam mass in the superheated region.

W_s = steam flow rate to turbine, which is an external constraint imposed by the controller.

W_b = steam production rate.

The heat balance equations of the two superheated steam nodes, node 1 and node 2, are given by:

$$\frac{d}{dt}(M_{s1} H_{s1} - P_s V_{s1}) = Q_{s1} + W_{21} H_{s2} - W_s H_{s1} - P_{s1} \dot{V}_{s1}$$

$$\frac{d}{dt}(M_{s2} H_{s2} - P_s V_{s2}) = Q_{s2} + W_b H_g - W_{21} H_{s2} - P_{s2} \dot{V}_{s2}$$

where

M_s = steam mass in the superheated region.

P_s = steam pressure in the superheated region.

V_s = steam volume in the superheated region.

H_s = specific enthalpy of the steam.

Q_{s1}, Q_{s2} = heat transfer rate to the two superheated nodes.

$$Q_{s1} = h_{ws} P_{ws} L_s (T_{w1} - T_{s1}) / 2$$

$$Q_{s2} = h_{ws} P_{ws} L_s (T_{w2} - T_{s2}) / 2$$

Assuming the pressure loss in the superheated steam region is small, we have:

$$P_s = P_{s1} = P_{s2}$$

Since specific enthalpy is a function of temperature and pressure, then we have:

$$\dot{H}_s = \frac{\partial H_s}{\partial T_s} \dot{T}_s + \frac{\partial H_s}{\partial P_s} \dot{P}_s$$

and

$$H_{s2} - H_{s1} = \frac{\partial H_s}{\partial T_s} (T_{s2} - T_{s1})$$

Combining with the mass balance equations and the expansion of the specific enthalpy, the energy balance equations can be rewritten as follows:

$$M_{s1} C_{ps} \dot{T}_{s1} + C_{ps} (T_{s1} - T_{s2}) \dot{M}_{s1} + (M_{s1} \frac{\partial H_s}{\partial P_s} - V_{s1}) \dot{P}_s = Q_{s1} - W_s C_{ps} (T_{s1} - T_{s2})$$

$$M_{s2} C_{ps} \dot{T}_{s2} + (M_{s2} \frac{\partial H_s}{\partial P_s} - V_{s2}) \dot{P}_s = Q_{s2} - W_b C_{ps} (T_{s2} - T_{sat})$$

The steam pressure in the superheated region can be described by compressibility adjusted ideal gas law, which is given by:

$$P_s V_s = Z^*_s M_s R (T_{s1} + T_{s2}) / (2 M_{stm})$$

The time derivative of the steam pressure can then be determined by the following equation:

$$\dot{P}_s = \left\{ \frac{Z^*_s R}{2 M_{stm}} \{ \dot{M}_s (T_{s1} + T_{s2}) + M_s (\dot{T}_{s1} + \dot{T}_{s2}) \} - P_s A_s \dot{L}_s \right\} \frac{1}{A_s L_s}$$

where

M_{stm} = mole mass of steam.

Z^*_s = steam expansion coefficient.

2.5 Equations for Boiling Region

The mass balance equation for the boiling region is given by:

$$\frac{d}{dt} (\bar{\rho}_b A_s L_b) = W_{db} - W_b$$

If we notice

$$\frac{d\bar{\rho}_b}{dt} = \frac{d\bar{\rho}_b}{dP_s} \frac{dP_{sat}}{dt}$$

then

$$A_s \bar{\rho}_b \dot{L}_b + A_s L_b K_b \dot{P}_{sat} = W_{db} - W_b$$

where

$$K_b = \frac{\partial \bar{\rho}_b}{\partial P}$$

If we assume that the steam quality is linear function of the axial position along the channel, then

$$\bar{\rho}_b = \frac{\int_0^1 \rho(x) dx}{\int_0^1 dx} = \int_0^1 \frac{dx}{v_f + xv_{fg}}$$

In the operation pressure range, we have:

$$\bar{\rho}_b(P_s) = 1.61594 + 0.00552445P_s$$

Therefore,

$$\dot{L}_b = \{W_{db} - W_b - A_s L_b K_b \dot{P}_{sat}\} / (A_s \bar{\rho}_b)$$

$$W_b = h_{wb} U_{wb} \left(\frac{T_{w3} + T_{w4}}{2} - T_{sat} \right) L_b / h_{fg}$$

W_{db} = flow rate leaving subcooled region to the saturated region.

h_{fg} = vaporization heat.

2.6 Equations for Subcooled Region

In analogy to the boiling region, the mass balance equation for the subcooled region can be given as follows:

$$W_{db} = W_{fw} - \bar{\rho}_{sc} A_{sc} \dot{L}_{sc} - A_s L_{sc} K_{sc} \dot{P}_{sc}$$

where

$$K_{sc} = \frac{\partial \bar{\rho}_{sc}}{\partial P_{sc}} = \frac{\partial}{\partial P_{sc}} (\rho_{fw} + \rho_f) / 2$$

W_{fw} = feed water flow rate.

Heat balance equation for the subcooled region 1 is given by:

$$\frac{d(MC_p T)_{sc1}}{dt} - V_{sc1} \dot{P}_{sc} = h_{wsc} P_{wsc} L_{sc} (T_{w5} - T_{sc1}) + C_{psc} (W_{sc} T_{sc2} - W_{db} T_{sc1})$$

Since the outlet temperature of the first subcooled node can be approximated by the saturated temperature, then

$$\begin{aligned} & \frac{A_s (\rho C_p)_{sc}}{2} [T_{sat} \dot{L}_{sc} + K_1 L_{sc} \dot{P}_{sat}] - \frac{A_s L_{sc}}{2} \dot{P}_{sc} \\ & = h_{wsc} P_{wsc} L_{sc} (T_{w5} - T_{sat}) / 2 + C_{psc} (W_{sc} T_{sc2} - W_{db} T_{sat}) \end{aligned}$$

where

$$K_1 = \frac{\partial T_{sat}}{\partial P}$$

Heat balance equation for the subcooled region 2:

$$\begin{aligned} & \frac{A_s(\rho C_p)_{sc}}{2} [T_{sc2} \dot{L}_{sc} + L_{sc} \dot{T}_{sc2}] - \frac{A_s L_{sc}}{2} \dot{P}_{sc} \\ & = h_{wsc} U_{wsc} L_{sc} (T_{w6} - T_{sc2}) / 2 + C_{psc} (W_{fw} T_{fw} - W_{sc} T_{sc2}) \end{aligned}$$

After simplification, we obtain

$$\begin{aligned} \dot{T}_{sc2} & = \{ \{ 0.5 * h_{wsc} P_{wsc} L_{sc} (T_{w6} - T_{sc2}) + C_{psc} (W_{fw} T_{fw} - W_{sc} T_{sc2}) \\ & + \frac{A_s L_{sc}}{2} \dot{P}_{sc} \} * 2 / (A_s(\rho C_p)_{sc} - T_{sc2} \dot{L}_{sc}) \} / L_{sc} \end{aligned}$$

If we assume $\dot{M}_{sc1} = \dot{M}_{sc2}$, then we have:

$$W_{sc} = (W_{fw} + W_{db}) / 2$$

Substituting the expression of W_{sc} and W_{db} into the heat balance equation for the subcooled region 1, we have:

$$\begin{aligned} \dot{L}_{sc} & = \frac{1}{0.5(A_s(\rho C_p)_{sc})(T_{sc2} - T_{sat})} \{ \{ 0.5 * h_{wsc} P_{wsc} L_{sc} (T_{w5} - T_{sat}) + C_{psc} (W_{fw} T_{sc2} - W_{sc} T_{sat}) \\ & - \frac{A_s L_{sc}}{2} [K_{sc} C_{psc} (T_{sc2} - 2T_{sat}) - 1] \dot{P}_{sc} \} - 0.5 A_s(\rho C_p)_{sc} K_1 L_{sc} \dot{P}_{sat} \} \end{aligned}$$

Noticing the pressure relationship between P_{sc} , P_{sat} and P_s , we have

$$P_{sc} = P_{sat} + \frac{1}{2}(\Delta P_{tpb} + \Delta P_{spsc})$$

$$P_{sat} = P_{ss} + \frac{1}{2}(\Delta P_{tpb} + \Delta P_{spss})$$

where

P_{sc} = pressure at the subcooled region.

P_{sat} = pressure at the saturated region.

ΔP_{tpb} = two phase pressure loss in the boiling region.

ΔP_{spsc} = single phase pressure loss in the subcooled region.

ΔP_{spss} = single phase pressure loss in the superheated region.

2.7 Equations for the Pressure Controller

The secondary side pressure is maintained by regulating the steam flow rate. The steam flow rate satisfies the following equation:

$$\dot{W}_s = \frac{W_{s0}(1 - C_{st}u) - W_s}{\tau_s}$$

where

u = controller output.

τ = time constant.

W_{s0} = initial steam flow rate on the secondary side.

C_{st} = an adjustable parameter.

If a PI controller is used, the controller output has both the proportional part $u_1(t)$ and the integral part $u_2(t)$, which is given by:

$$u_1(t) = k_1 \left(\frac{P_{tb}}{P_0} - \frac{P_{set}}{P_0} \right)$$

$$\frac{du_2(t)}{dt} = k_2 \left(\frac{P_{tb}}{P_0} - \frac{P_{set}}{P_0} \right)$$

where

k_1 = proportional gain.

k_2 = integral gain.

P_{tb} = turbine header pressure.

P_{set} = turbine header pressure set point.

P_0 = turbine header pressure set point.

3. Steam Turbine

The model development of the turbine system incorporates the following assumptions:

Reversible adiabatic expansion process exists in the nozzle chest and moisture separator.

Thermodynamic properties of the saturated steam and water are assumed to be linear functions of the steam pressure at each state.

The bypass flow to the reheater is proportional to the main steam pressure and steam valve coefficient.

Heat transfer coefficients for the reheater and feedwater heaters are assumed to be constant.

Steam flow losses in the turbine system are considered to be zero.

To simplify the model complexity, modifications have been made for the evaluation of high pressure and low pressure turbine efficiencies. The efficiencies of the high and low pressure turbines are considered to be constant within a range of ± 100 PSI deviations in the exhaust pressure of the high pressure turbine from its initial value (at full power of operating condition of the plant).

$$\eta_{hp} = \frac{h_c - h_2}{h_c - h_2'} \\ \eta_{lp} = \frac{h_r - h_4}{h_r - h_4'}$$

Variable	Definition
η_{hp}	High pressure turbine efficiency
η_{lp}	Low pressure turbine efficiency
h_2, h_4	Steam enthalpy leaving the high and low pressure turbine at points 2 and 4 respectively
h'_2, h'_4	Isotropic enthalpy at points 2 and 4
h_c	Steam enthalpy leaving the nozzle chest

4. Main Condenser

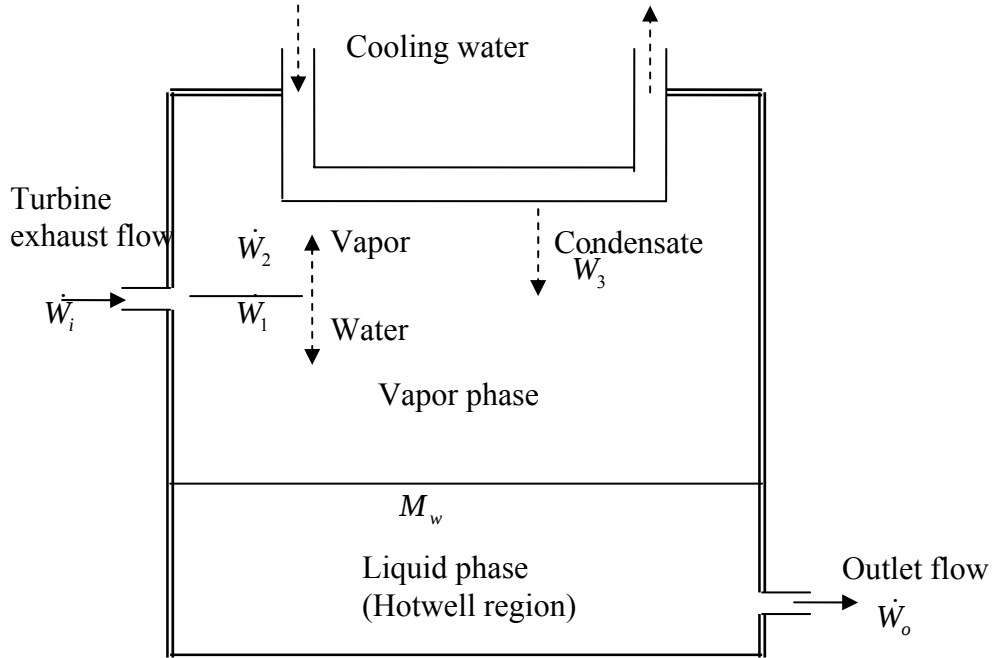
A condenser is a large surface-type heat exchanger, which condenses steam from the exhaust of turbine by transferring steam latent heat to circulating water inside the condenser. The condenser is desired to work under vacuum condition to obtain a maximum mechanical power from the turbine system.

The main condenser is simulated as an equilibrium tow-phase tank. The space inside the tank is divided into tow independent control volumes, steam and water. They are assumed to be in thermal equilibrium during steady state condition. Turbine exhaust flow enters the system at the condenser pressure. The water part of the flow falls into the hotwell region and mixes perfectly with the water already present there. The vapor part condenses on the outer surface of metal tubes through which the circulating water flows. The condensation process is associated with a time delay which is due to the dynamics of heat transfer between the vapor and the circulating water. The following assumptions are made to simplify the mathematical model.

Constant pressure exists in the condenser model.

Mass transfer at steam-liquid interface is assumed to be negligible.

Wall condensation is considered to be zero.



The describing equations of the model are derived based on the conservation of mass and energy equations at each phase. These equations are introduced below.

Mass Balance Equations

Liquid phase:

$$\frac{dM_w}{dt} = \dot{W}_1 + \dot{W}_3 - \dot{W}_o$$

Vapor phase:

$$\frac{d\dot{W}_3}{dt} = \frac{\dot{W}_2 - \dot{W}_3}{\tau_\infty}$$

Energy Equations

$$\begin{aligned} \frac{d}{dt}(M_w h_o) &= (\dot{W}_1 + \dot{W}_3) h_f - \dot{W}_o h_o \\ \frac{dh_o}{dt} &= \frac{(\dot{W}_1 + \dot{W}_3)(h_f - h_o)}{M_w} \end{aligned}$$

Constitutive Relations

Mass balance:

$$\dot{W}_1 = \dot{W}_i - \dot{W}_2$$

where

$$\dot{W}_2 = \dot{W}_i \frac{(h_i - h_f)}{h_{fg}}$$

Variable	Definition
M_w	Water mass inside the condenser
$\dot{W}_1$	Water droplet rate into the hotwells
$\dot{W}_3$	Water condensation flow rate
$\dot{W}_o$	Outlet flow rate to the low pressure feed water heater
$\dot{W}_2$	Steam flow rate in the condenser
τ_{co}	Time constant of the condensation process
h_f	Enthalpy of the saturated water
h_o	Enthalpy of the outlet water
$\dot{W}_i$	Entering flow rate to the condenser
h_{fg}	Latent enthalpy of the water
h_i	Enthalpy of the entering flow rate

5. Feedwater Heater

Low Pressure Feedwater Heater Equations

Energy balance:

$$\frac{dh_{fw1}}{dt} = \frac{Q_{h1}}{\tau_{h1}W_{fw}} + \frac{h_o - h_{fw1}}{\tau_{h1}}$$

Heat flow relation:

$$Q_{h1} = H_{fw1}(W_{hp1} + W_{b1p})$$

High Pressure Feedwater Heater Equations

Mass balance:

$$\frac{dW_{hp1}}{dt} = \frac{W_{bhp} + W_{ms} + W_{pr1} - W_{hp1}}{\tau_{h2}}$$

Energy balance:

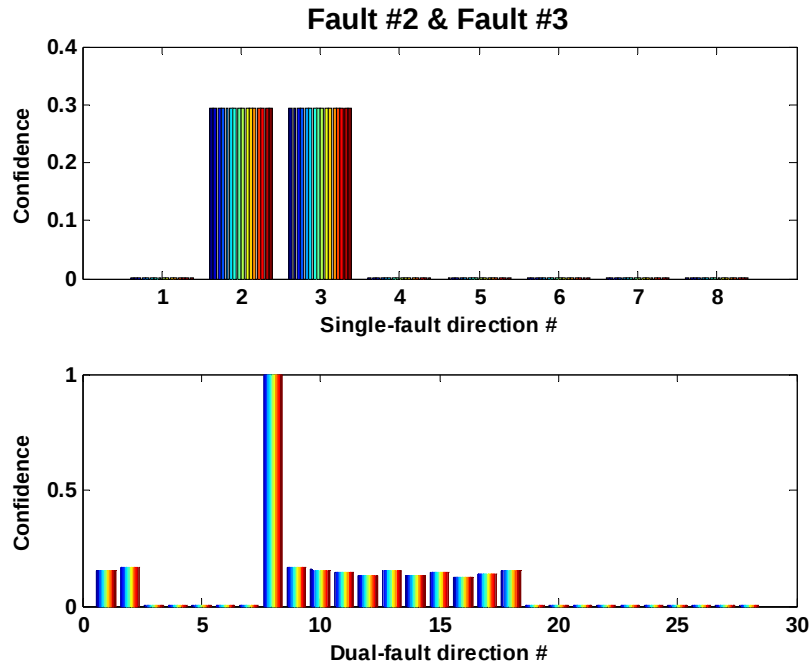
$$\frac{dh_{fw2}}{dt} = \frac{Q_{h2}}{\tau_{h2}W_{fw}} + \frac{h_{fw1} - h_{fw2}}{\tau_{h2}}$$

Heat flow relation:

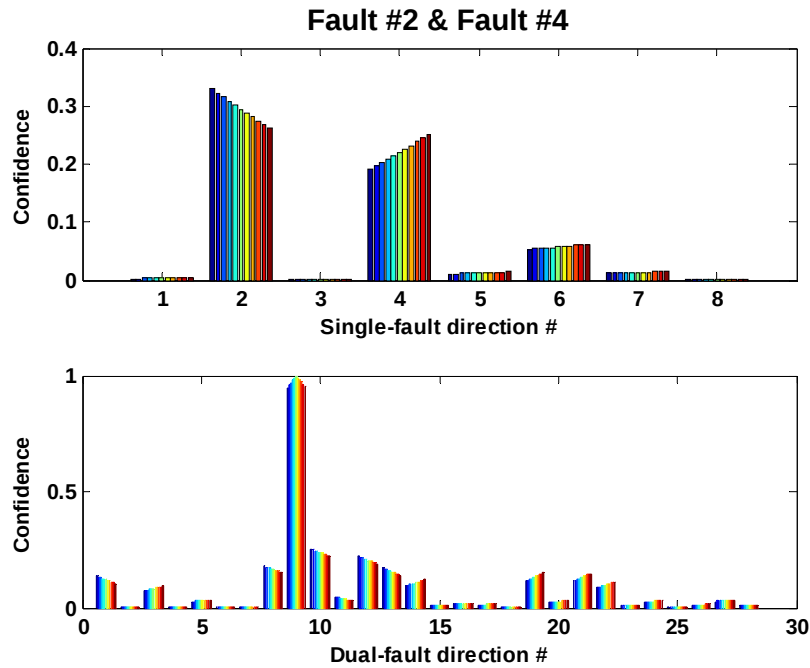
$$Q_{h2} = H_{fw2}(W_{ms} + W_{pr1} + W_{bhp})$$

Appendix C: Fault Isolation Index Plots of Dual-Fault Scenarios

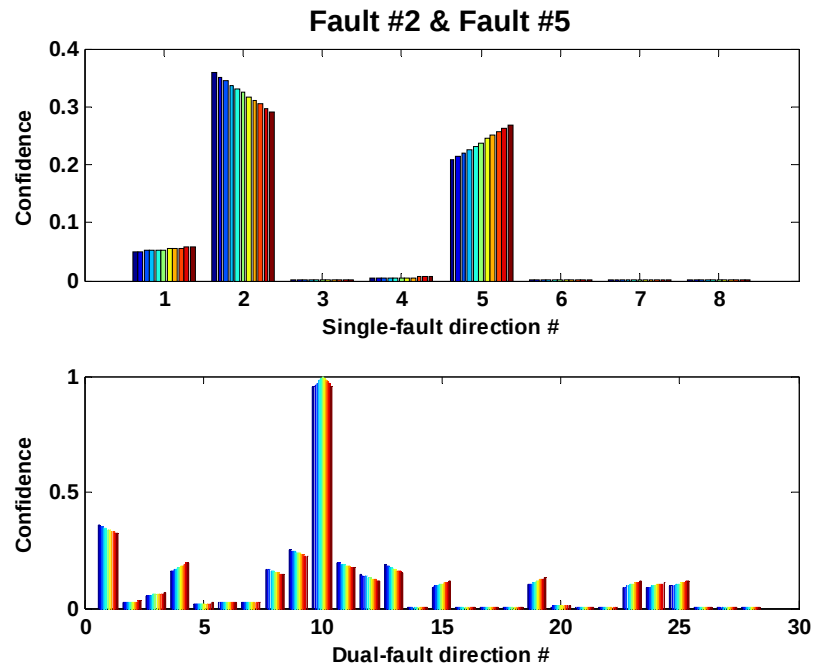
Dual-fault case #8:



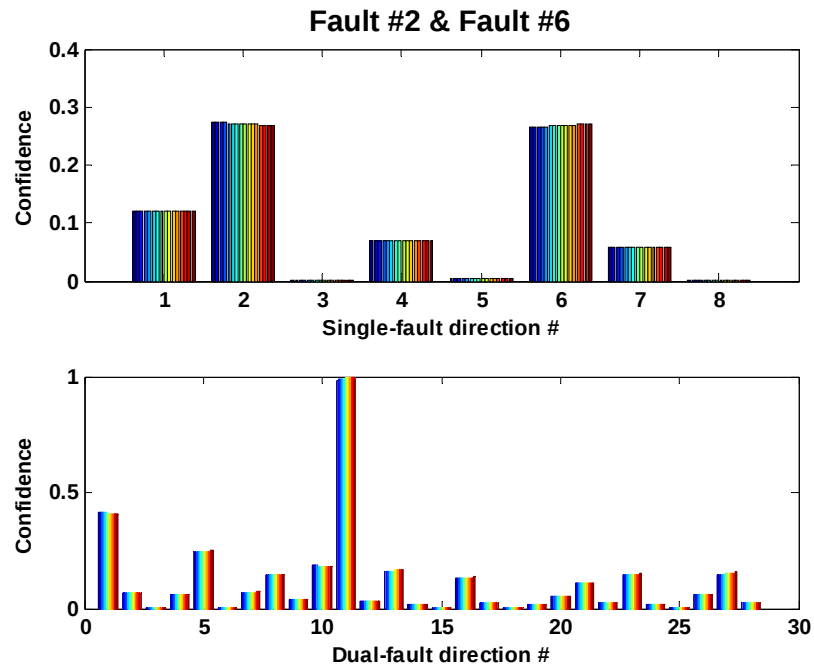
Dual-fault case #9:



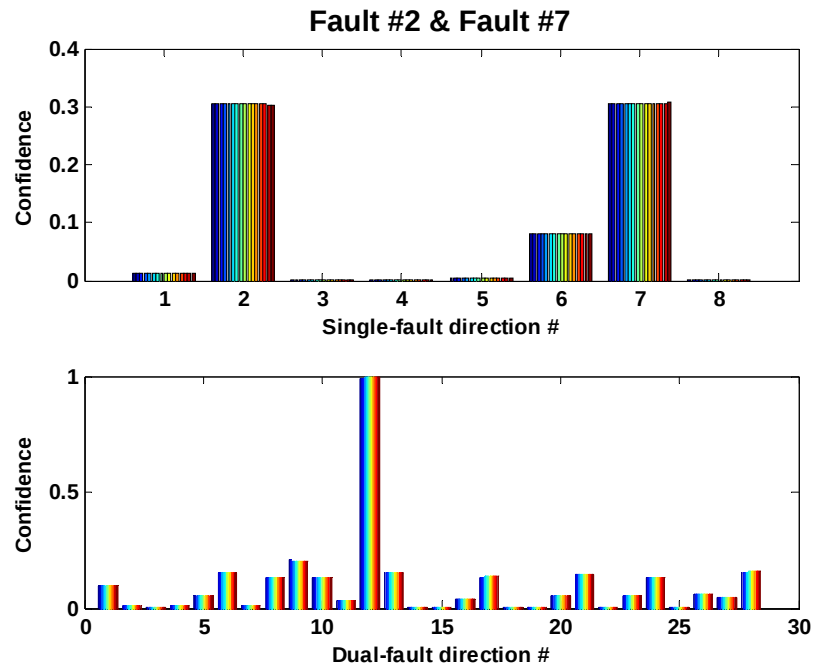
Dual-fault case #10:



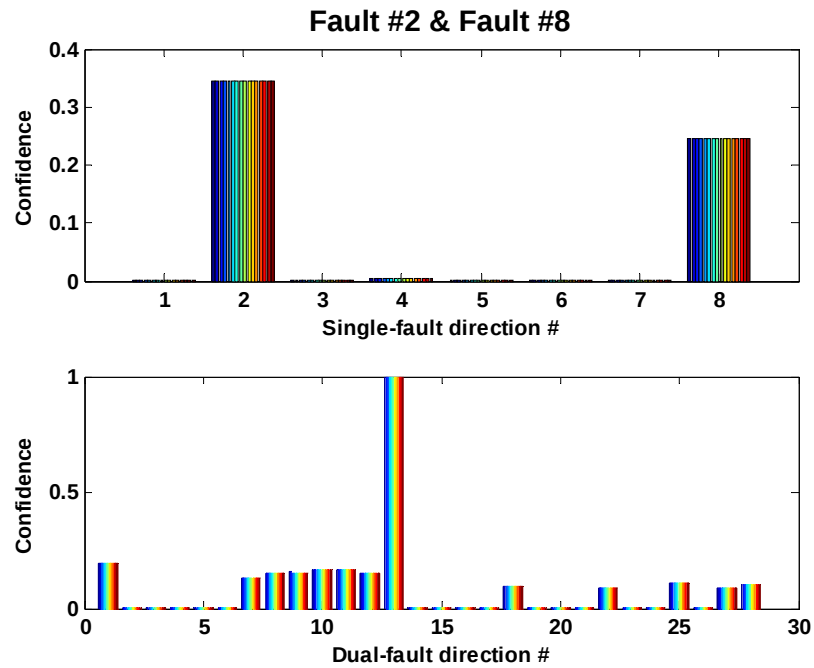
Dual-fault case #11:



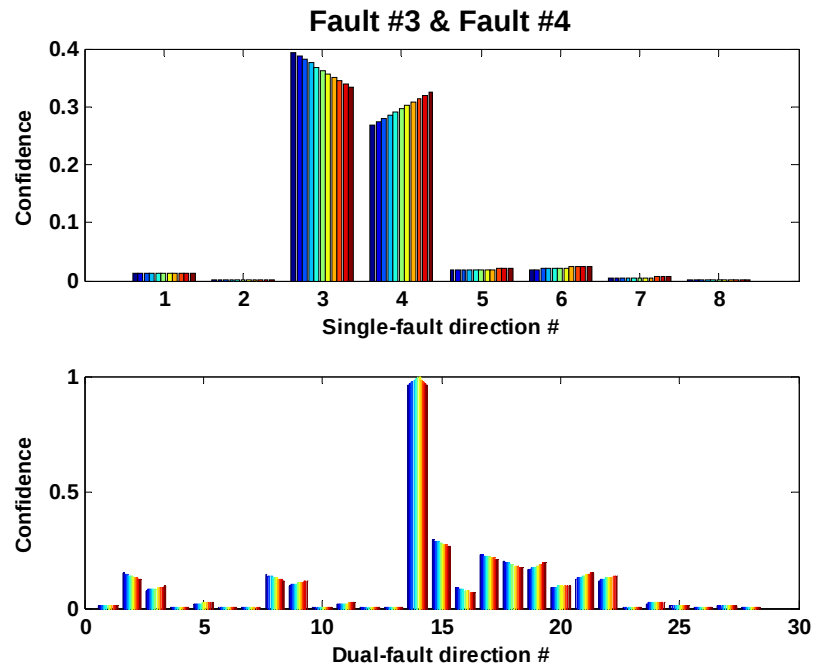
Dual-fault case #12:



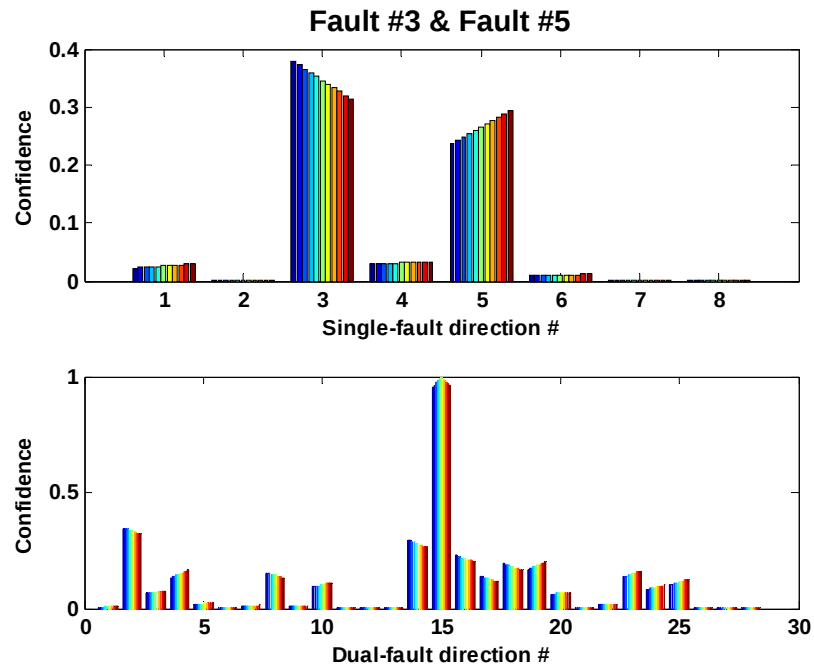
Dual-fault case #13:



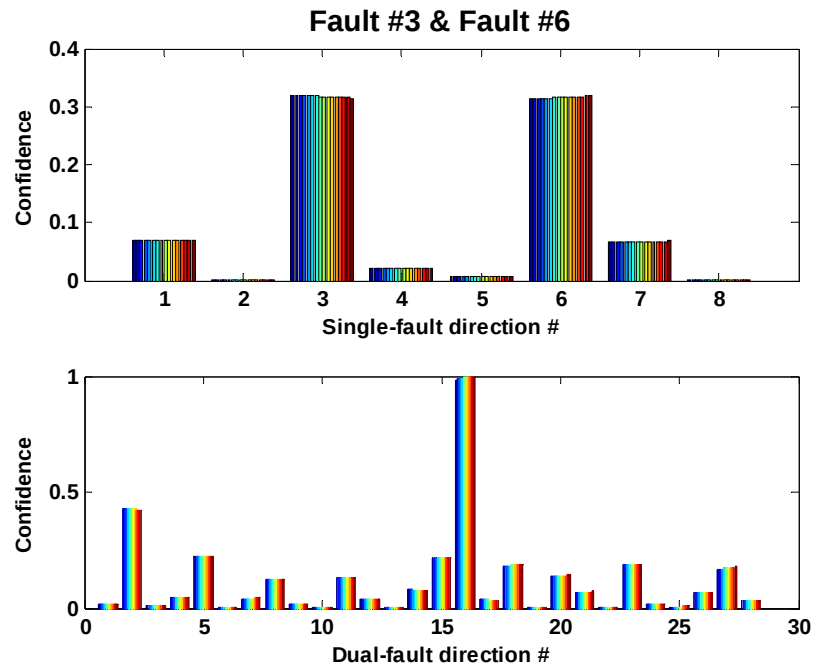
Dual-fault case #14:



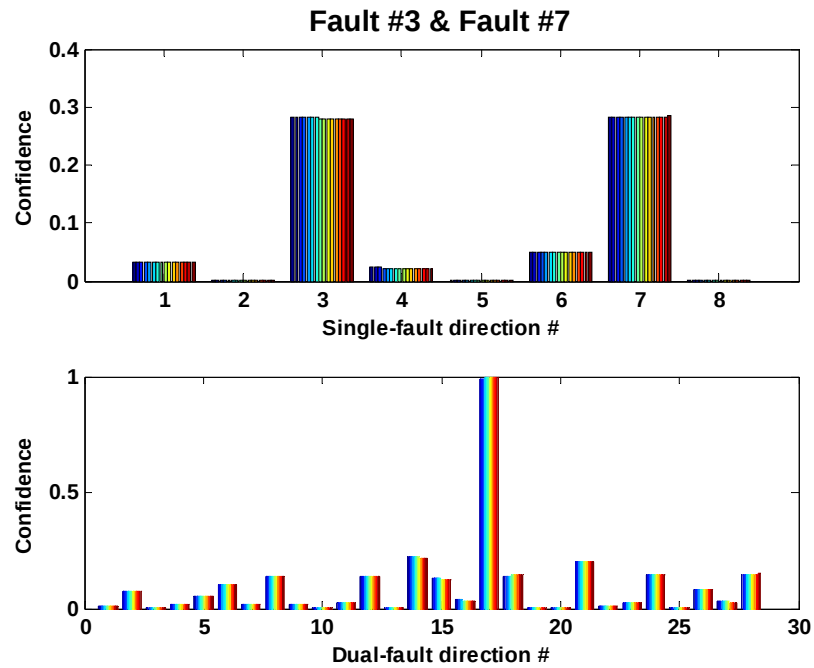
Dual-fault case #15:



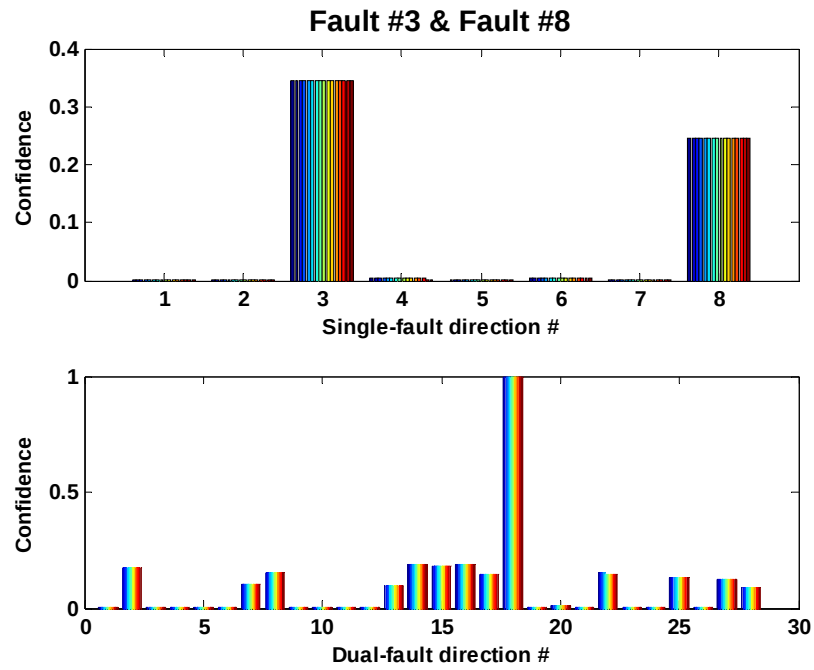
Dual-fault case #16:



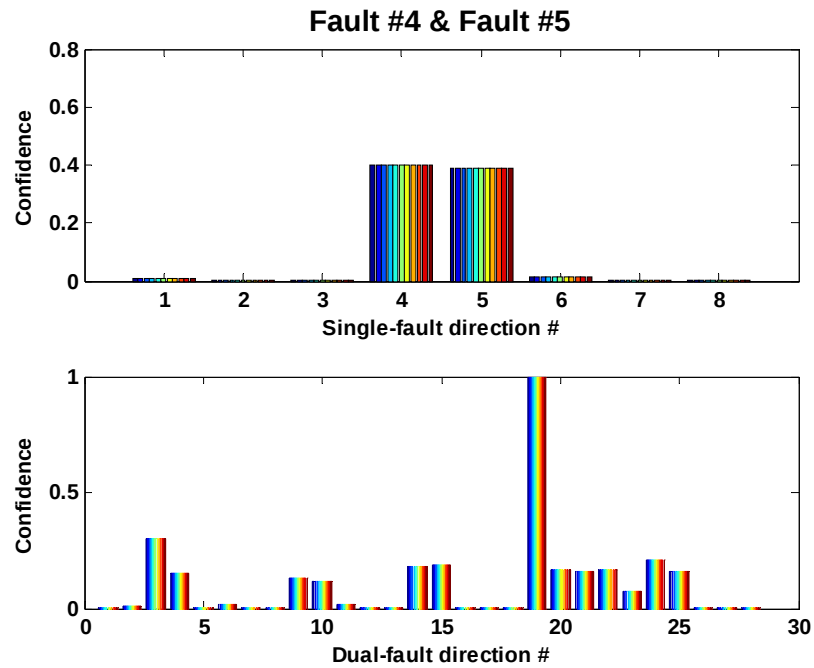
Dual-fault case #17:



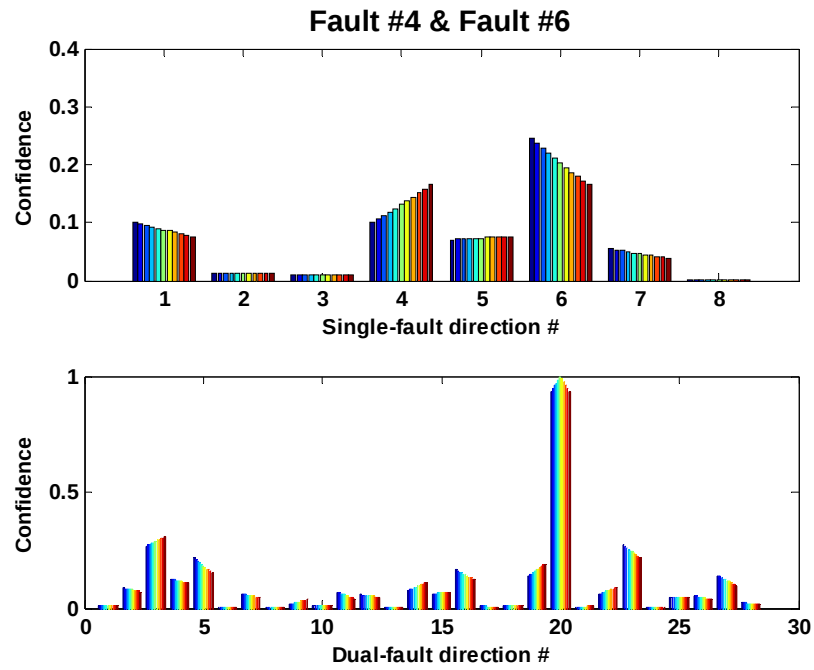
Dual-fault case #18:



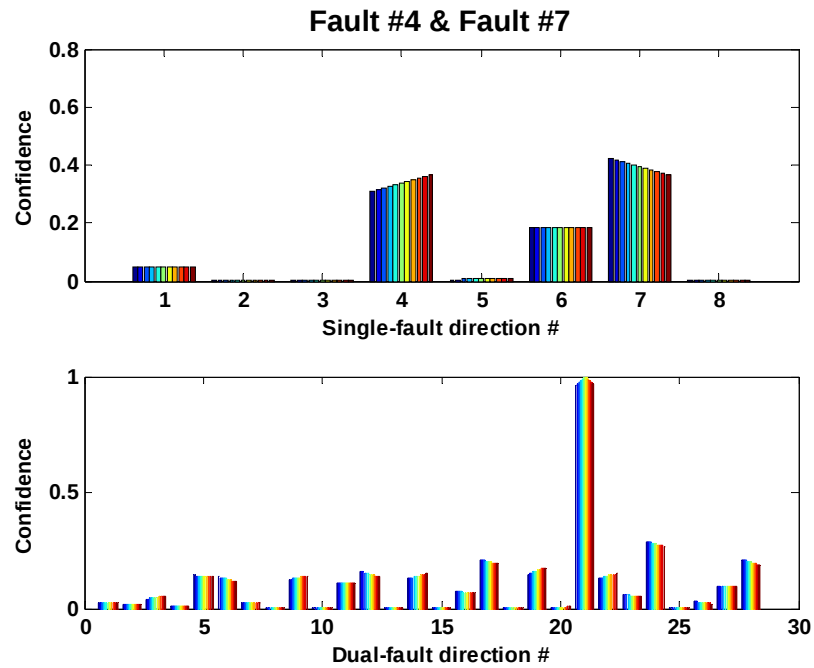
Dual-fault case #19:



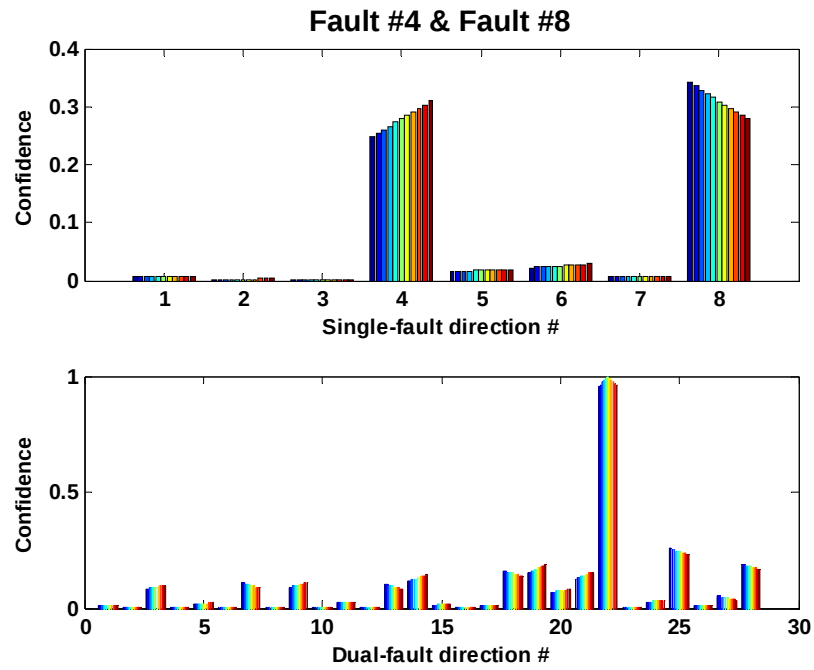
Dual-fault case #20:



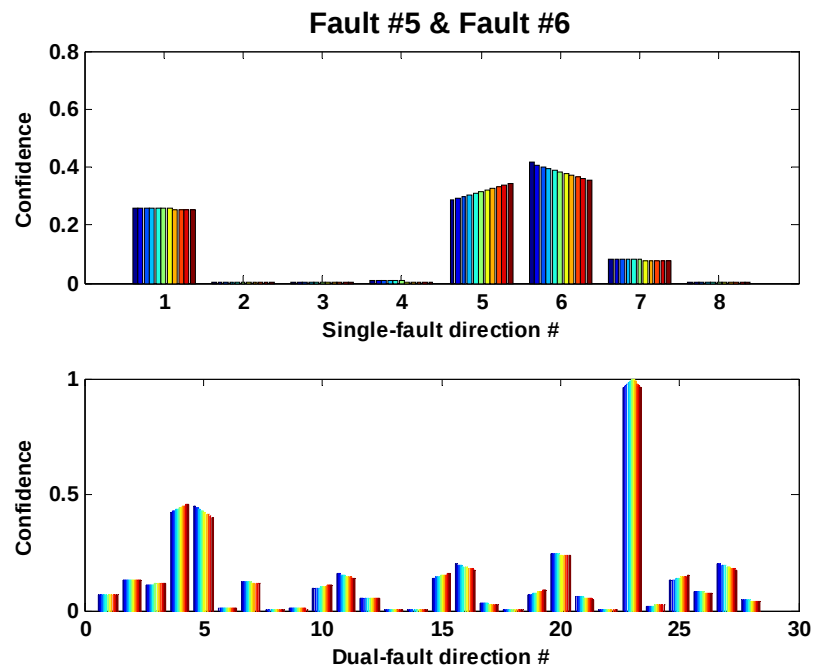
Dual-fault case #21:



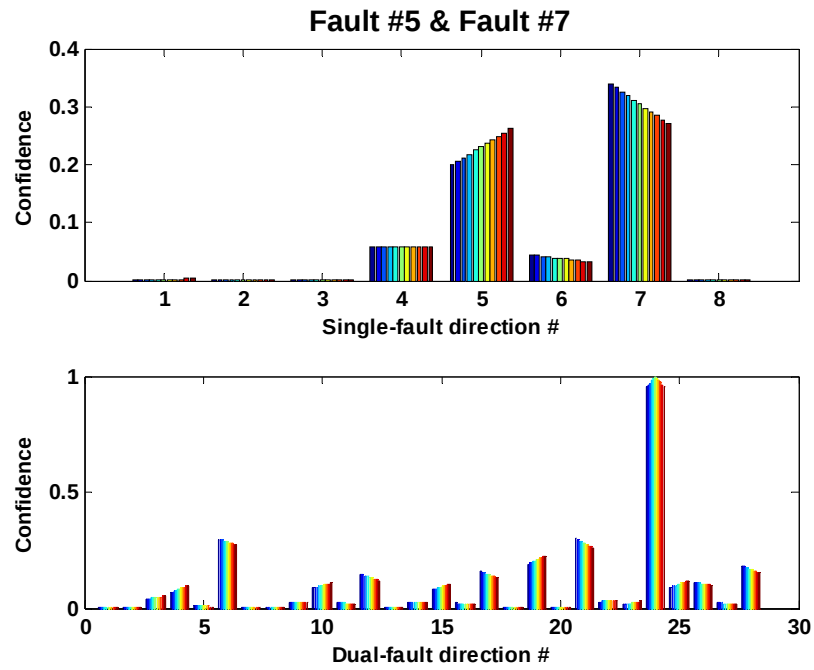
Dual-fault case #22:



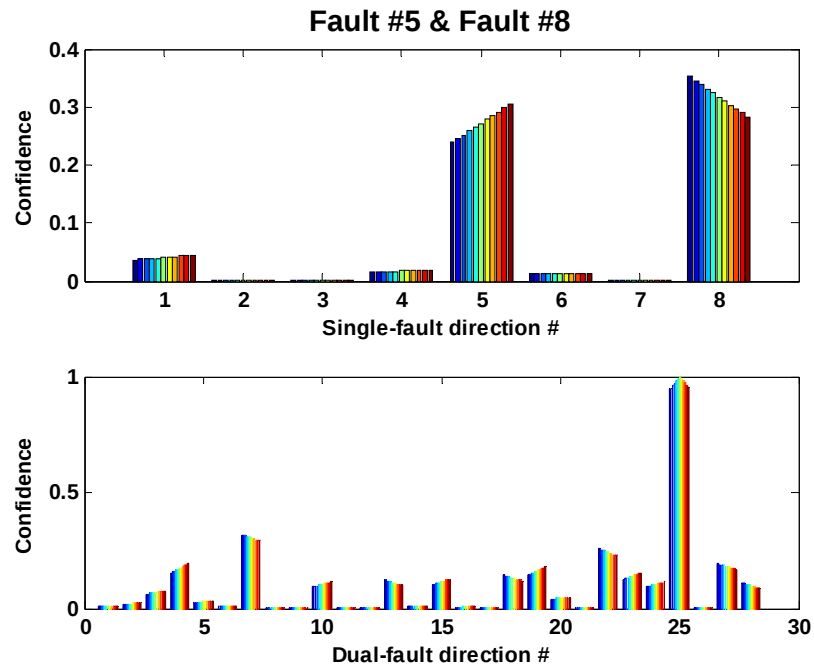
Dual-fault case #23:



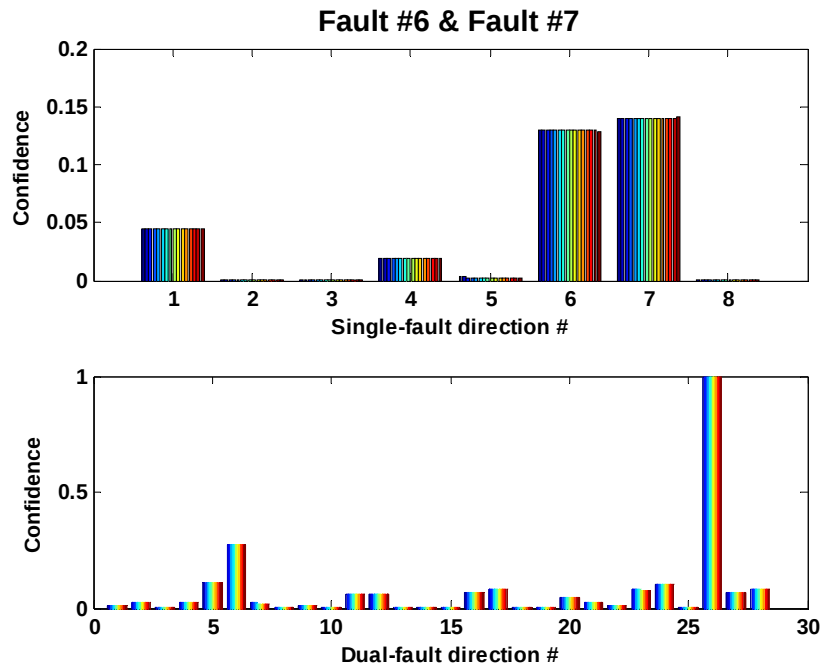
Dual-fault case #24:



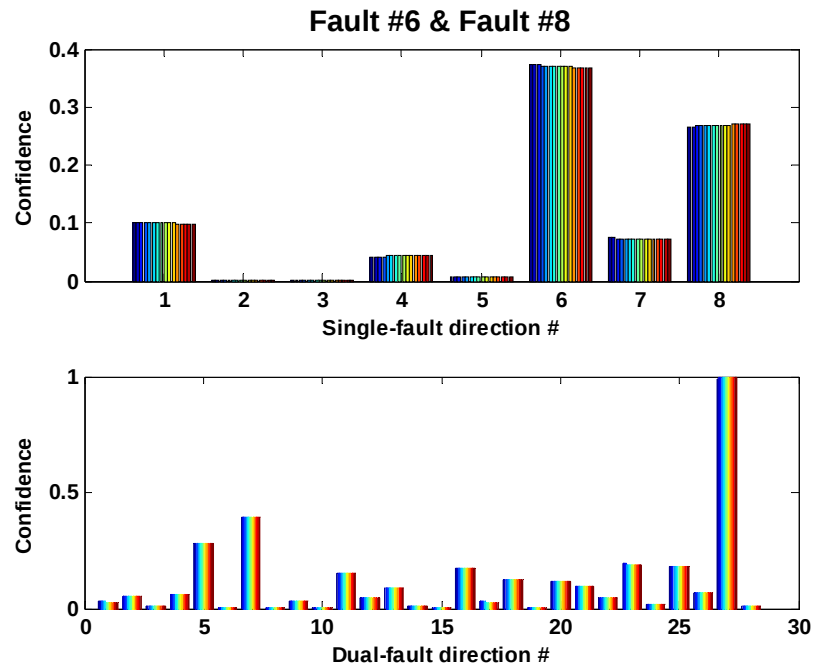
Dual-fault case #25:



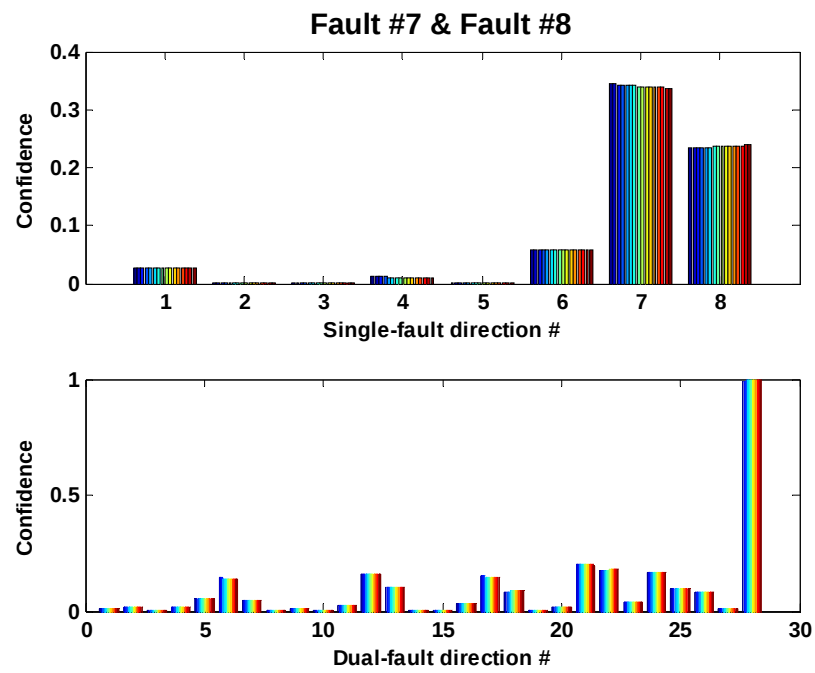
Dual-fault case #26:



Dual-fault case #27:



Dual-fault case #28:



Appendix D: MSF Desalination Plant Simulation Codes

Steady-state Simulation Code :

```
% The steady state MSF simulation
clear all;
close all;
clc

% Heat recovery section tube inner diameter (in.)
di_rec = 0.866;
% Heat recovery section tube thickness (in.)
th_rec = 0.048;
% Heat recovery section tube length (ft)
len_rec = 40;

% Heat rejection section tube inner diameter (in.)
di_rej = 0.9394;
% Heat rejection section tube thickness (in.)
th_rej = 0.0303;
% Heat rejection section tube length (ft)
len_rej = 35;

% Brine heater tube inner diameter (in.)
di_br = 0.866;
% Brine heater tube thickness (in.)
th_br = 0.048;
% Brine heater tube length (ft)
len_br = 40;

% Seawater temperature (F)
Tsea = 95;
% Cooling seawater mass flowrate (lb/hr)
Lsea = 0.249*1.0E8;
% Seawater salt concentration (weight percentage)
Xsea = 5.6;

% Recycle stream flowrate (lb/hr)
R = 0.140*1.0E8;
% Make-up flowrate (lb/hr)
F = 0.125*1.0E8;

% Reject cooling seawater flowrate (lb/hr)
Cw = Lsea-F;

%

% Heat transfer area of heat recovery stages (ft^2)
A_rec = 43000;
% Heat transfer area of heat rejection stages (ft^2)
A_rej = 38000;
% Heat transfer area of brine heater (ft^2)
A_br = 38000;
```

```

% Steam temperature (F)
Ts = 206;
% Reference temperature (F)
Tref = 32;
% Height of brine pool in each stage (in.)
H = 18;
% Width of a recovery stage (ft)
w_rec = 40;
% Width of a rejection stage (ft)
w_rej = 35;

% Initialize all the variables, assuming 3 stages and a brine heater
% Temperatures (F): cooling brine temperatures Tw(1-3)
%           flashing brine temperature Tb(1-3)
%           distillate product saturation temperature Tp(1-3)
% Tw = [192;184;176];
% Tb = [192.66;186;179.47];
% Tp = [190.53;183.66;177.29];

Tw = [190;183;176;169;162;155;148;141;134;127;120;113;110;110;103;100;98];
Tb = [190;183;176;169;162;155;148;141;134;127;120;113;110;110;103;100;98];
Tp = [190;183;176;169;162;155;148;141;134;127;120;113;110;110;103;100;98];

% Flowrates (lb/hr):
% Lw: cooling brine mass flowrates in the recovery section
% Lb: flashing brine mass flowrates
% Lp: distillate product mass flowrates
Lw_rec = R+F;
Lw_rej = Lsea;
%Lb = [0.2633;0.26165;0.2600]*1.0E8;
%Lp = [0.17;0.34;0.51]*1.0E6;

% Salt concentration (mass fraction): cooling brine concentration Xw
%           flashing brine concentrations Xb(1-3)
Xw_rej = Xsea;
Xb = [6.4;6.45;6.5;6.55;6.6;6.65;6.7;6.75;6.8;6.85;6.9;6.95;7;7;7.05;7.1;7.15];

% brine heater outlet flow properties
Lb0 = Lw_rec;
Tb0 = 200;
Xb0 = 6.25;

count = 0;
Tb_ref = zeros(17,1);
while sum((Tb-Tb_ref).^2)>17*1.0E-5

Tb_ref = Tb;

Cpb0 = heatcapa(Tb0,Xb0);
hb0 = hbrine(Tb0,Xb0);
% Specific enthalpy of flashing brine leaving each stage
for i = 1:17
    hb(i) = hbrine(Tb(i),Xb(i));

```

```

end

% Specific enthalpy of flashing vapor at each stage
for i = 1:17
    hv(i) = hsteamsat(Tp(i));
end

% step 1 in the loop
Lb(1) = Lb0*(hb0-hv(1))/(hb(1)-hv(1));
for i = 2:13
    Lb(i) = Lb(i-1)*(hb(i-1)-hv(i))/(hb(i)-hv(i));
end
Lb(14) = Lb(13);
for i = 15:17
    Lb(i) = Lb(i-1)*(hb(i-1)-hv(i))/(hb(i)-hv(i));
end
% step 2 in the loop
Lp(1) = Lb0-Lb(1);
for i = 2:13
    Lp(i) = Lb(i-1)+Lp(i-1)-Lb(i);
end
Lp(14) = Lp(13);
for i = 15:17
    Lp(i) = Lb(i-1)+Lp(i-1)-Lb(i);
end
% Blowdown mass flowrate (lb/hr)
BD = Lb(17)-R;

% Salt balance on mixer for Xw_rec
Xr = Xb(17);Xf = Xw_rej;
Xw_rec = (R*Xr+F*Xf)/Lw_rec;
% Salt concentration of brine heater outlet flow
Xb0 = Xw_rec;

% step 5 in the loop
Xb(1) = Lb0*Xb0/Lb(1);
for i = 2:13
    Xb(i) = Lb(i-1)*Xb(i-1)/Lb(i);
end
Xb(14) = Xb(13);
for i = 15:17
    Xb(i) = Lb(i-1)*Xb(i-1)/Lb(i);
end

% The overall heat transfer coefficients for each of the stages
for i = 1:13 % recovery stages
    U(i) = htcoeff(Tp(i),Tw(i),Lw_rec,Xw_rec,A_rec,di_rec,th_rec,len_rec);
end
U(14) = 0;
for i = 15:17 % rejection stages
    U(i) = htcoeff(Tp(i),Tw(i),Lw_rej,Xw_rej,A_rej,di_rej,th_rej,len_rej);
end

```

```

% Specific heat capacity of cooling brine leaving each stage
for i = 1:13 % recovery stages
    Cpw(i) = heatcapa(Tw(i),Xw_rec);
end
Cpw(14) = heatcapa(Tw(14),Xw_rec);
for i = 15:17 % rejection stages
    Cpw(i) = heatcapa(Tw(i),Xw_rej);
end

Cpsea = heatcapa(Tsea,Xsea);
% Mean specific heat capacity of cooling brine at each stage
for i = 1:13
    Cpav(i) = (Cpw(i)+Cpw(i+1))/2;
end
Cpav(14) = 0;
for i = 15:16
    Cpav(i) = (Cpw(i)+Cpw(i+1))/2;
end
Cpav(17) = (Cpw(17)+Cpsea)/2;

% Specific heat capacity of distillation product leaving each stage
for i = 1:17
    Cpp(i) = heatcapa(Tp(i),0);
end
% Specific heat capacity of flashing brine leaving each stage
for i = 1:17
    Cpb(i) = heatcapa(Tb(i),Xb(i));
end

% alpha calculation for each stage
for i = 1:13
    alpha(i) = exp(U(i)*A_rec/Lw_rec/Cpav(i));
end
alpha(14) = 0;
for i = 15:17
    alpha(i) = exp(U(i)*A_rej/Lw_rej/Cpav(i));
end

% parameter calculation for each stage
a1(1) = 0;
for i = 2:13
    a1(i) = Lp(i-1)*Cpp(i-1)/Lw_rec/Cpav(i);
end
a1(14) = a1(13);
for i = 15:17
    a1(i) = Lp(i-1)*Cpp(i-1)/Lw_rej/Cpav(i);
end

a2(1) = Lb0*Cpb0/Lw_rec/Cpav(1);
for i = 2:13
    a2(i) = Lb(i-1)*Cpb(i-1)/Lw_rec/Cpav(i);
end
a2(14) = a2(13);

```

```

for i = 15:17
    a2(i) = Lb(i-1)*Cpb(i-1)/Lw_rej/Cpav(i);
end

for i = 1:13
    a3(i) = Lp(i)*Cpp(i)/Lw_rec/Cpav(i);
end
a3(14) = a3(13);
for i = 15:17
    a3(i) = Lp(i)*Cpp(i)/Lw_rej/Cpav(i);
end

for i = 1:13
    a4(i) = Lb(i)*Cpb(i)/Lw_rec/Cpav(i);
end
a4(14) = a4(13);
for i = 15:17
    a4(i) = Lb(i)*Cpb(i)/Lw_rej/Cpav(i);
end

for i = 1:17
    a5(i) = (a3(i)+a4(i)-a1(i)-a2(i))*Tref;
end

Z(1) = bpe(Tb(1),Xb(1))+nea(Tb0,Tb(1),H,w_rec,Lw_rec,Tp(1))+temploss(Tp(1));
for i = 2:13
    Z(i) =
    bpe(Tb(i),Xb(i))+nea(Tb(i-1),Tb(i),H,w_rec,Lw_rec,Tp(i))+temploss(Tp(i));
end
Z(14) = Z(13);
for i = 15:17
    Z(i) =
    bpe(Tb(i),Xb(i))+nea(Tb(i-1),Tb(i),H,w_rej,Lw_rej,Tp(i))+temploss(Tp(i));
end

for i = 1:17
    b1(i) = a1(i)+a2(i);
    b2(i) = -a3(i)-a4(i);
end

b3(1) = a5(1)+a2(1)*Tb0-a4(1)*Z(1);
for i = 2:17
    b3(i) = a5(i)+a2(i)*Z(i-1)-a4(i)*Z(i);
end

c1(1) = 0; %
for i = 2:17
    c1(i) = b1(i)*alpha(i-1)/(alpha(i-1)-1);
end

c2(1) = b2(1)*alpha(1)/(alpha(1)-1);
for i = 2:17
    c2(i) = b1(i)/(1-alpha(i-1))-b2(i)*alpha(i)/(1-alpha(i));

```

```

end

for i = 1:17
    c3(i) = b2(i)/(1-alpha(i));
end

% Rejection section cooling brine temperatures
TDM_rej = zeros(3,3);
Y_rej = zeros(3,1);
TDM_rej(1,1:3) = [-1-alpha(15)*b2(15)/(1-alpha(15)) c3(15)+1 0];
TDM_rej(2,1:3) = [c1(16) c2(16)-1 c3(16)+1];
TDM_rej(3,1:3) = [0 c1(17) c2(17)-1];

Y_rej(1) = -b3(15)-b1(15)*(Tw(14)-alpha(13)*Tw(13))/(1-alpha(13));
Y_rej(2) = -b3(16);
Y_rej(3) = -b3(17)-Tsea*(c3(17)+1);

Tw(15:17) = inv(TDM_rej)*Y_rej;

% Mixer enthalpy balance
Tw(14) = (R*Cpb(17)*(Tb(17)-Tref)+F*Cpw(15)*(Tw(15)-Tref))/Lw_rec/Cpw(14)+Tref;

% Recovery section cooling brine temperatures
TDM_rec = zeros(13,13);
Y_rec = zeros(13,1);
TDM_rec(1,1:2) = [c2(1)-1 c3(1)+1];
for i = 2:12
    TDM_rec(i,i-1:i+1) = [c1(i) c2(i)-1 c3(i)+1];
end
TDM_rec(13,12:13) = [c1(13) c2(13)-1];
for i = 1:12
    Y_rec(i) = -b3(i);
end
Y_rec(13) = -b3(13)-Tw(14)*(c3(13)+1);

Tw(1:13) = inv(TDM_rec)*Y_rec;

% Update thermal parameters
% The overall heat transfer coefficients for each of stages
for i = 1:13 % recovery stages
    U(i) = htcoeff(Tp(i),Tw(i),Lw_rec,Xw_rec,A_rec,di_rec,th_rec,len_rec);
end
U(14) = 0;
for i = 15:17 % rejection stages
    U(i) = htcoeff(Tp(i),Tw(i),Lw_rej,Xw_rej,A_rej,di_rej,th_rej,len_rej);
end

% Specific heat capacity of cooling brine leaving each stage
for i = 1:13 % recovery stages
    Cpw(i) = heatcapa(Tw(i),Xw_rec);
end
Cpw(14) = heatcapa(Tw(14),Xw_rec);
for i = 15:17 % rejection stages

```

```

    Cpw(i) = heatcapa(Tw(i),Xw_rej);
end

Cpsea = heatcapa(Tsea,Xsea);
% Mean specific heat capacity of cooling brine at each stage
for i = 1:13
    Cpav(i) = (Cpw(i)+Cpw(i+1))/2;
end
Cpav(14) = 0;
for i = 15:16
    Cpav(i) = (Cpw(i)+Cpw(i+1))/2;
end
Cpav(17) = (Cpw(17)+Cpsea)/2;

% alpha calculation for each stage
for i = 1:13
    alpha(i) = exp(U(i)*A_rec/Lw_rec/Cpav(i));
end
alpha(14) = alpha(13);
for i = 15:17
    alpha(i) = exp(U(i)*A_rej/Lw_rej/Cpav(i));
end

% step 7 in the loop
% obtain distillation product temperature for each stage
for i = 1:13
    Tp(i) = (Tw(i+1)-alpha(i)*Tw(i))/(1-alpha(i));
end
Tp(14)=Tp(13);
for i = 15:16
    Tp(i) = (Tw(i+1)-alpha(i)*Tw(i))/(1-alpha(i));
end
Tp(17) = (Tsea-alpha(17)*Tw(17))/(1-alpha(17));

% update Z
Z(1) = bpe(Tb(1),Xb(1))+nea(Tb0,Tb(1),H,w_rec,Lw_rec,Tp(1))+temploss(Tp(1));
for i = 2:13
    Z(i) =
    bpe(Tb(i),Xb(i))+nea(Tb(i-1),Tb(i),H,w_rec,Lw_rec,Tp(i))+temploss(Tp(i));
end
Z(14) = Z(13);
for i = 15:17
    Z(i) =
    bpe(Tb(i),Xb(i))+nea(Tb(i-1),Tb(i),H,w_rej,Lw_rej,Tp(i))+temploss(Tp(i));
end

% step 8 in the loop
% obtain flashing brine temperature for each stage
for i = 1:17
    Tb(i) = Tp(i)+Z(i);
end

% calculate the maximum brine temperature Tb0
U_br = htcoeff(Ts,Tb0,Lw_rec,Xw_rec,A_br,di_br,th_br,len_br); % heat transfer

```

```

coefficient of brine heater
Cpav_br = (Cpw(1)+Cpb0)/2;
alpha_br = exp(U_br*A_br/Lw_rec/Cpav_br); % alpha calculation for brine heater
Tb0 = (Tw(1)-(1-alpha_br)*Ts)/alpha_br;
count = count+1;
end
Tb0
Ws = Lb0*Cpav_br*(Tb0-Tw(1))/(hsteamsat(Ts)-hwatersat(Ts))
Tb
Tp
Tw
Count

```

Dynamic Simulation Code :

```
% The dynamic MSF simulation
close all;
clear all;
clc

% Heat recovery section tube inner diameter (in.)
di = 0.866;
% Heat recovery section tube thickness (in.)
th = 0.048;
% Heat recovery section tube length (ft)
len = 40;

Axs = 36; % chamber area (m^2)
Apxs = 3; % distillation tray area (m^2)

w = 9.6; % orifice width (m)
h = 0.11; % orifice height (m)
C = 0.623; % orifice contraction coefficient
%K = 0.6413; % orifice discharge coefficient
K1 = 0.6467;
K2 = 0.6319;

wp = 0.6; % orifice width (m)
hp = 0.11; % orifice height (m)
Cp = 0.623; % orifice contraction coefficient
%Kp = 0.1235; % orifice discharge coefficient
Kp1 = 0.0615;
Kp2 = 0.1214;
% specific heat capacity (Btu/lb°F)
Cpb0 = 0.9486;
Cpb1 = 0.9472;
Cpb2 = 0.9457;
Cpp1 = 1.0041;
Cpp2 = 1.0033;
Cpw1 = 0.9454; % Btu/lb°F
Cpw2 = 0.9443; % Btu/lb°F

U1 = 676.7345/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
U2 = 674.91/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
A = 43000; % heat transfer area (ft^2)
U_br = 696.745/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
A_br = 38000; % heat transfer area (ft^2)

rhow = 63.1691; %cooling brine density (lb/ft^3)
rhob = 1010.6561133; %flashing brine density in kg/m^3
rhop = 965; % distillate product density in kg/m^3

% surface area per tube (ft^2)
As = 2*pi*((di/2+th)/12)*len;
% # of tubes in one stage
tube = A/As;
```

```

% # of tubes in brine heater
tube_br = A_br/As;
% volume per tube (ft^3)
Vol = 0.25*pi*(di/12)^2*len;

Mw = rhow*Vol*tube; % total mass in tubes (lb)
Mb0 = rhow*Vol*tube_br; % total mass in tubes (lb)

% inputs
Tw_in = 178.55;
Lw_in = 0.265*1.0E8/3600; % lb/sec
Xw_in = 6.2412;

Tv1=192.5;
Tv2=186.7;
Hfg1 = hsteamsat(Tv1)-hwatersat(Tv1); % vapor latent heat (Btu/lb)
Hfg2 = hsteamsat(Tv2)-hwatersat(Tv2);
Pv1 = pft(Tv1);
Pv2 = pft(Tv2);
Pv_in = 0.5328; % 3rd stage pressure value in bar

Tsteam = 206;
Lsteam = 73.6; % lb/sec

Kv1 = 0.0599;
Kv2 = 0.058;

% controller parameters
TBTsetpoint = 200;
Kp=.1;Ki=.1;
Levelsetpoint = 0.6264;
Kp_level=.01;Ki_level=.0005;

```

Appendix E: IRIS-MSF Nuclear Desalination Simulation Codes

```
% The IRIS-MSF nuclear desalination simulation
close all;
clear all;
clc

% Heat recovery section tube inner diameter (in.)
di = 0.866;
% Heat recovery section tube thickness (in.)
th = 0.048;
% Heat recovery section tube length (ft)
len = 40;

Axs = 36; % chamber area (m^2)
Apxs = 3; % distillation tray area (m^2)

w = 9.6; % orifice width (m)
h = 0.11; % orifice height (m)
C = 0.623; % orifice contraction coefficient
%K = 0.6413; % orifice discharge coefficient
K1 = 0.6467;
K2 = 0.6319;

wp = 0.6; % orifice width (m)
h_p = 0.11; % orifice height (m)
Cp = 0.623; % orifice contraction coefficient
%Kp = 0.1235; % orifice discharge coefficient
Kp1 = 0.0615;
Kp2 = 0.1214;
% specific heat capacity (Btu/lb°F)
Cpb0 = 0.9486;
Cpb1 = 0.9472;
Cpb2 = 0.9457;
Cpp1 = 1.0041;
Cpp2 = 1.0033;
Cpw1 = 0.9454; % Btu/lb°F
Cpw2 = 0.9443; % Btu/lb°F

U1 = 676.7345/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
U2 = 674.91/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
A = 43000; % heat transfer area (ft^2)
U_br = 696.745/3600; % overall heat transfer coefficient Btu/(sec F ft^2)
A_br = 38000; % heat transfer area (ft^2)

rho_w = 63.1691; %cooling brine density (lb/ft^3)
rho_b = 1010.6561133; %flashing brine density in kg/m^3
rho_p = 965; % distillate product density in kg/m^3

% surface area per tube (ft^2)
A_s = 2*pi*((di/2+th)/12)*len;
% # of tubes in one stage
```

```

tube = A/A_s;
% # of tubes in brine heater
tube_br = A_br/A_s;
% volume per tube (ft^3)
Vol = 0.25*pi*(di/12)^2*len;

Mw = rho_w*Vol*tube; % total mass in tubes (lb)
Mb0 = rho_w*Vol*tube_br; % total mass in tubes (lb)

% inputs
Tw_in = 178.55;
Lw_in = 0.265*1.0E8/3600; % lb/sec
Xw_in = 6.2412;

Tv1=192.5;
Tv2=186.7;
Hfg1 = hsteamsat(Tv1)-hwatersat(Tv1); % vapor latent heat (Btu/lb)
Hfg2 = hsteamsat(Tv2)-hwatersat(Tv2);
Pv1 = pft(Tv1);
Pv2 = pft(Tv2);
Pv_in = 0.5328; % 3rd stage pressure value in bar

% Tsteam = 206;
% Lsteam = 73.6;

Kv1 = 0.0599;
Kv2 = 0.058;

% controller parameters
TBTsetpoint = 200.5;
Kp=.1;Ki=.1;
Levelsetpoint = 0.6264;
Kp_level=.01;Ki_level=.0005;

TwBias = 0.0;
LwBias = 0.0;
XwBias = 0.0;
Tb0Bias = 0.0;
TBTsetpointBias = 0.0;
LevelsetpointBias = 0.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% IRIS Reactor System Parameters

%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reactor Core
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Units converter
kk=1.054e-3; % 1BTU =1.054e3 J = 1.054 e-3MJ

%from mm to ft;

c1=0.1*0.3937/12;

%conversion from kg to lbm;

c2=1/0.4536; % 1lb= 0.4536kg

% 1 ft =0.3m

% One group delayed neutron fraction
% beta=0.0044;
% decay constant (sec-1)
% lamda=0.07561;

% 6 groups delayed neutron
beta = [0.000215,0.001424,0.001274,0.002568,0.000748,0.000273]; % Neutron
fraction

lamda =[0.0124,0.0305,0.111,0.301,1.14,3.01]; % Decay constant (sec-1)
%lamda =[0.0125,0.0308,0.1140,0.307,1.19,3.19];

beta_tot=sum(beta);

%C0=[2.56044980203996, 5.98380618825848, 1.50806782543339, ...
% 1.22026479381736, 0.09438247614957, 0.00809051173591]*1.0E11;
%C0= [2.3235,6.2567,1.5381,1.1433,0.0879,0.0122]*1.0e11;

% average prompt neutron generation time (s)
kapa=1.0e-4;

C0=lamda.\beta/kapa;

% Nominal Power output (MWth)
P0=1000;

% total coolant mass flow rate
% Core flow rate (lbm/sec)
Wc=3.7397e7/3600;
%Wc=3.7e7/3600;
W=Wc/8; % Coolant going to one HCSG

% Mann's two nodes model
Wc1=Wc;

```

```

Wc2=Wc;

mf = 1.8e5; % Mass of fuel (lb)

% total mass of coolant in core
mc= 7.45e5; % lb

mc1=mc/2;
mc2=mc-mc1;

%% reactor core thermal model
% Moderator temperature coefficient of reactivity (1/F)
Alpha_c = -4.0e-4;
%Alpha_c = -3.6e-4 ; % 1/K;
% Fuel Temperature Coefficient of Reactivity (1/F)
Alpha_f = -1.65e-5; % 1/K

% Fraction of Total Power Deposited in Fuel
Fc = 0.97;

% Specific Heat capacity of Fule (BTU/lbm-F)
Cpf = 0.059*kk;
% Cpf=246.5; % (J/kg-K)
% Specific heat capacity of Moderator (BTU/lbm-F)
Cpc = 1.394*kk;
%Cpc = 5807; % (J/kg-K)
% Effective fuel-to-coolant heat transfer area (ft2)
Afc = 41631.6;
Afc1=Afc/2;
Afc2=Afc/2;
%Afc =5570.7; % m2

% overall fule-to-coolant heat transfer coefficient (BTU/Hr-ft2-F)
Ufc = 327.4/3600*kk;
%Ufc = 5.672; % J/s-kg-K
rho_h2o = 45.71; %lbm/ft3
%rho_h2o = 732.6 ; % kg/m3

% lower riser volume
Vlr=19.0*35.288; %ft3
% upper riser volume
Vur = 35.4 * 35.288; %ft3
%RCP suction plenum
Vcp = 23.5 *35.288; %ft3

% 1 m3 = 35.288 ft3
Vtotal=(19.0+35.4+23.5)/8*35.288; % ft3
Vhl=1000;

mm=rho_h2o*Vtotal;

%Tfuel0=841.59;
Tfuel0=841.62;

```

```

%Theta10=(557.6+(622.4-557.6)/3-32)*5/9;
Theta10=590.65;
%Theta20=(557.6+(622.4-557.6)*2/3-32)*5/9;
Theta20=622.4;
%Theta20=623.12;
Thot0=624;
%Thot0=623.12;
%Tcold0=557.6;
%
Tcold0=556;
sim('coress_MSF');

%P0 = core_ss(end,3);
%C0 = core_ss (end,4:9);
%Tfuel0 = core_ss(end,10);
%Theta10 = core_ss(end,11);
%Theta20 = core_ss(end,12);
%Thot0 = core_ss(end,13);

%ini_core=[P0,C0,Tfuel0,Theta10,Theta20,Thot0];
ini_core =core_ss(end,3:end);
coreSS=core_ss(end,:);
Kp_shim=0.1;%0.0001;
%Ki_shim=0.000044;
Kp_tavg=0.01;
Ki_tavg=0.00044;

Tavg_set=590.00;

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   IRIS HCSG Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

JJ=778.16; %BTU/ft/lbf;

Fudge1=1.0;
Fudge2=1.0;
%from mm to ft;
c1=0.1*0.3937/12;
%conversion from kg to lbm;
c2=1/0.4536;

%N =636;
N=656;
Ntube=N;
% Tube inner diameter=13.24mm
Ri_thermal=13.24/2*c1;
% Tube outside diameter=17.46mm
Ro_thermal=17.46/2*c1;
% Inlet temperature on the primary side

```

```

Tpin=ini_core(11);
%Tpin=1.8*328.62+32;%(From NCSU code: 328.62F, original=328.4F)
%Tube inside flow area;
%Ri_hydraulic=Ri_thermal;
%As=pi*Ri_hydraulic^2;

%Internal shell external diameter Di=0.61 m;
Di=0.61*1000*c1;
%External shell internal diameter Dt=1.62m
Do=1.62*1000*c1;
%Radial pitch=25mm;
t=25*c1;
%Total shell side projected area
Ap_total=1/4*pi*(Do^2-Di^2)*(1-Ro_thermal^2/t);
%Shell side hydraulic radius=2*flow area/wetting perimeter
%Wetting area
Peri_wet=N*2*pi*Ro_thermal;
Ro_hydraulic=2*Ap_total/Peri_wet;
Ap=pi*Ro_hydraulic^2;

%Tube side hydraulic diameter
Ri_hydraulic=Ri_thermal;
%Flow area on the secondary side;
As=pi*(Ri_hydraulic^2);
%Cross section for the tube;
Aw=pi*(Ro_thermal^2-Ri_thermal^2);

%Specific heat capacity;
%Cpp=1.3355;
Cpp=1.39;
Cpw=0.109;
%Cps=1.0185;
Cpfw=1.122;
Cpsc=1.122;

%Density lbm/ft**3
rho_p=44.75;
rho_w=526.0;
%rhos=1.876;
rhoavg=8.86; %for entire boiling region;
rhofw=51.71;
rhof=46.91; %for boiling water
rhosc=(rhofw+rhof)/2;

Ps=5.8;
%Ps=Ps;
Ts=(280+318)/2+273;
[dum,dum,Cps,vss]=hsh(Ps,Ts);
Cps=Cps*9.4783E-4/(2.2046*1.8);
rhos=1/vss*0.06243;

%Initial heat transfer coefficient

```

```

%hpw=2212/3600;
%hws=710/3600;
%hwsc=1758/3600;
%hwb=6000/3600;

Rii=Ri_thermal;
Roo=Ro_thermal;
ccc=0.1761;

Kw=(10.1924/3600)*1.0;
hp=21000*ccc/3600.0;
hs=8500*ccc/3600.0;
hsc=16000*ccc/3600.0;
hb=55000*ccc/3600.0;

hpw=Kw*hp/(Kw+hp*Roo*(0.5+log(Roo/Rii)/(1-(Roo/Rii)^2)));
hws=Kw*hs/(Kw+hs*Rii*(log(Rii/Roo)/(((Rii/Roo)^2)-1)-0.5));
hwsc=Kw*hsc/(Kw+hsc*Rii*(log(Rii/Roo)/(((Rii/Roo)^2)-1)-0.5));
hwb=Kw*hb/(Kw+hb*Rii*(log(Rii/Roo)/(((Rii/Roo)^2)-1)-0.5));

%Feed water temperature=224C;
Tfw=1.8*224+32;

%Tube length
Lt=32*3.2808; %total bundle length
Lb=21.5*3.2808; %this value is fixed to determine accurate heat transfer
coefficient in this region.
Lsc=4.5*3.2808; %this value is malipulated given that hwsc is known.
Ls=Lt-Lb-Lsc;
%%%%%Heating circumference%%%%%
Ri=Ri_thermal;
%Ro=Ro_thermal;
Uwb=2*pi*Ri;
Uws=2*pi*Ri;
Uwsc=2*pi*Ri;

Ptable=5.0:0.1:6.0;
Tsavg=(280+318)/2+273;
Ttable=[];
HfgTable=[];
hsTable=[];
for PPP=5.0:0.1:6.0
[Tsat,hf,hg,kf,kg,muf,mug,Prf,Prg]=hsat(PPP);
[dum,hss,dum,dum]=hsh(PPP,Tsavg);
Ttable=[Ttable,Tsat];
hsTable=[hsTable,hss];
HfgTable=[HfgTable,hg-hf];
end;
c3=1000/6.895;
Ptable=Ptable*c3; %Pressure;
Ttable=(Ttable-273)*1.8+32; %Saturated temperature;
cc1=9.4783E-4/2.2046;

```

```

HfgTable=HfgTable*cc1;
hsTable=hsTable*cc1;
a=polyfit(Ptable,Ttable,1);
X5=a(2);K5=a(1);
b=polyfit(Ptable,HfgTable,1);
X4=b(2); K4=b(1);
c=polyfit(Ptable,hsTable,1);
dHsdPs=c(1);

N0=8;
%Flow rate on the primary side=589 kg/s; 1298.5 lb/s (per SG);
%Core flow rate=589*8=4712 kg/s
Wp=4712*c2/N/N0*Fudge1;
Wp0=Wp;
Wp1=Wp;
Wp2=Wp;
Wp3=Wp;
Wp4=Wp;
Wp5=Wp;

Fraction=1.0;
%Flow rate on the secondary side=62.85kg/s (per SG)
%Total secondary flow=62.85*8=502.8kg/s

AdjustFactor=1.0;
Wsec=502.8*c2/N0/N*Fraction*AdjustFactor;
Ws=Wsec;
Wb=Wsec;
Wfw=Wsec;
Wsg=Wsec;
Ws0=Wsec;
Ri=Ri_thermal;
Ro=Ro_thermal;

%Preparing data matrix;
a1=Ap*Cpp*rhop/2; %primary side;
a2=Aw*Cpw*rhow/2; %metal;
a3=As*Cps*rhos/2; %secondary side;
a4=hpw*pi*Ro/a1;

a51=Cpp*Wp0/a1;
a52=Cpp*Wp1/a1;
a53=Cpp*Wp2/a1;
a54=Cpp*Wp3/a1;
a55=Cpp*Wp4/a1;
a56=Cpp*Wp5/a1;

a6=hpw*pi*Ro/a2;
a7=hwb*pi*Ri/a2;
a8=hws*pi*Ri/a2;
a9=hws*pi*Ri/a3;

```

```

a11=As*Cpsc*rhosc/2; %
a12=hwsc*pi*Ri/a2; %
a13=hwsc*pi*Ri/a11; %
a14=144/(JJ*Cps*rhop);
%dHsdPs=(1245.9-1251.8)/50.0;
a15=144/(JJ*Cps*rhos)-dHsdPs/Cps;
dHsdPsc=(430.47-430.19)/500;
a16=144/(JJ*Cpsc*rhosc)-dHsdPsc/rhosc;

%Saturated temperature for 7Mpa
c3=1000/6.895; % 1Mpa=1000/6.895 psi
Ps=5.8*c3;
%Ps=862;
deltaP=0.2;
%Psat=(5.8+deltaP)*c3;
Psat=Ps;
%X5=402.94; K5=0.14; %Tsat~Psat
Tsat=X5+K5*Psat;
%Tsat=546.6; %Exit temperature=317C and Degree of superheat is 43.4;
Hfg=X4+K4*Psat;
sim('hcs_gss_MSF');
a99=hws*pi*Ri/Cps;
a88=2*Wfw/As/rhosc;
a77=2/As/rhosc;
Ksc=(1/0.02152-1/0.02145)/20;
Kb=0.00552445;
dTsatsdP=K5;

Z=0.76634; % 570K, 60atm; steam expansion coefficient
R=4.55465*3.5314455E-5*14.696006/(5/9*2.2046223E-3); %cm^3atm/deg*g, gas
constant
Ct=Ws/Ps;
Ktb=1.5428e-004;

% Steam valve position controller
Pset=5.8*c3; % 5.8Mpa -> psi
%Pset= 862;

% Adjustable parameters
cst=10;
%cst=0.1;
taos=1; % the smaller, the faster response, more oscillation
%taos=0.5;

ti=10;
%ti2=1; %unit 2
kc=.5; % 0.5 is the original value change the response of pressure; the smaller,
the larger change

PsBias=0.0;
WsBias=0.0;

% sim('HCSG')

```

```

% ini_hcsg=out_hcsg(end,:);
%
%
% %%% IRIS Primary
%
% sim('Primary_IRIS')
%
% coreSS=core_tran(end,:);
% HCSGSSV=para_out(end,:);

%% Balance of Plant
epsilon = 0.6354;
Kapp = 3.4137e6; % 1MW = 3.4137e6 BTU/hour
% number of steam generators
Nsg = 8;
gc=32.174;
% turbine control valve open loss coefficient
Ktcv=44.85;
Refload = 335; % MWe

Wfw = 4.9877e5/3600; % Feedwater flow rate at 100% power level per SG (lbm/sec)

%% Condenser
K1co = 37.74;
K2co = 32;
K5co = 1054;
K6co = -18;
Pco0 = 0.71; % Condenser pressure (psi)
TOUco = 7;
MLco = 4.1423e4;

K_msf = 0.081;
K_fwh = 0.2;
TOU = 10;

%% Feedwater heater
TOUh = 20;
CPfw = 1.2181;

```

Appendix F: Greedy Search Optimization Algorithm Codes

Fault Observability:

```
% greedy search algorithm
% under single-fault assumption
clc
clear
% Taking number of root nodes from user
n=input('Please enter the number of root nodes = ');

% Taking number of key components from user
m=input('Please enter the number of key components = ');

con=zeros(m,n);
for p=1:m
    for q=1:n
        con(p,q)=input(['Arcs between key components and root nodes
[C',num2str(p),'],'[R',num2str(q),'] = ']);
    end
end

% combine the sensors that are connected to the same root node set
row = [];
for i = m:-1:2
    if nnz(con(i,:))==0
        continue
    end
    record = [];
    for j = i-1:-1:1
        if nnz(con(j,:))==0
            continue
        end
        sumcol = sum(con([i j],:),1); % add two rows up
        if nnz(sumcol==1)==0 % identical rows consisting of 0 and 2
            if nnz(i==record)==0 % no i in the record
                record = [record;i];
            end
            if nnz(j==record)==0
                record = [record;j];
            end
        end
    end
    if isempty(record)~=1 % if record is not empty
        last = record(size(record,1));
        if nnz(last==row)==0
            row = [row;last];
        end
    else % if record is empty
        if nnz(i==row)==0
            row = [row;i];
        end
    end
end
```

```

end
if (nnz(1==row)==0) && (nnz(con(1,:))>0)
    row = [row;1];
end
% array "row" consists of sensors that have different root node set.

% remove redundant sensors
for i=1:m
    if nnz(i==row)==0
        con(i,:)=0;
    end
end

% combine the root nodes that are connected to the same sensor set
column = [];
for i = 1:n-1
    record = [];
    for j = i+1:n
        sumcol = sum(con(:,[i j]),2); % add two columns up
        if nnz(sumcol==1)==0 % identical columns consisting of 0 and 2
            if nnz(i==record)==0 % no i in the record
                record = [record i];
            end
            if nnz(j==record)==0
                record = [record j];
            end
        end
    end
    if isempty(record)~=1 % if record is not empty
        last = record(size(record,2));
        if nnz(last==column)==0
            column = [column last];
        end
    else % if record is empty
        if nnz(i==column)==0
            column = [column i];
        end
    end
    if i==n-1
        if nnz(i+1==column)==0
            column = [column i+1];
        end
    end
end
% array "column" consists of root nodes that have different sensor set.

con = con(:,column); % combine indistinguishable faults
[m,n] = size(con);

% creating set B for all columns of the new connectivity matrix for single
% fault isolation
B = [];
for i=1:n-1
    for j=i+1:n

```

```

        sumcol = sum(con(:,[i j]),2);
        sumcol(find(sumcol==2))==0;
        B = [B sumcol];
    end
end
con = [con B];
[m,n] = size(con);

%original greedy search algorithm to find the minimal set of sensors for
observability
con_backup = con;

count=1;
C=[0];
while nnz(con)>0
for i=1:m
    if nnz(i==C)>0 % i is in C
        arc(i)=0;
        continue
    else % only select unmarked C with highest unmarked arcs (denoted "1")
        arc(i)=sum(con(i,find(con(i,:)==1)),2);
    end
end
% arc_1=sum(con,2);
maxarc=find(arc==max(arc));
% dealing with single unmarked C or multiple unmarked C
% mark and store C
if size(maxarc,2)==1
    count=count+1;
    C(count)=maxarc;
else
    % combine the identical sensors
    markedcon = zeros(m,n);

    for i=1:m
        if nnz(i==maxarc)>0 % i is in maxarc
            markedcon(i,find(con(i,:)==1))=1;
        end
    end

    row2 = [];
    for i = m:-1:2
        if nnz(markedcon(i,:))==0
            continue
        end
    end
    record = [];
    for j = i-1:-1:1
        if nnz(markedcon(j,:))==0
            continue
        end
        sumcol = sum(markedcon([i j],:),1); % add two rows up
        if nnz(sumcol==1)==0 % identical rows consisting of 0 and 2
            if nnz(i==record)==0 % no i in the record
                record = [record;i];
            end
        end
    end
end

```

```

        end
        if nnz(j==record)==0
            record = [record;j];
        end
    end
end
if isempty(record)~=1 % if record is not empty
    last = record(size(record,1));
    if nnz(last==row2)==0
        row2 = [row2;last];
    end
else % if record is empty
    if nnz(i==row2)==0
        row2 = [row2;i];
    end
end
end
end
if (nnz(markedcon(1,:))>0) && (nnz(1==row2)==0)
    row2 = [row2;1];
end

maxarc = sort(row2'); % new maxarc
count=count+size(maxarc,2);
C(count-size(maxarc,2)+1:count)=maxarc;
end

% generate all the root nodes covered by C, store them in matrix deletedarc
deletedarc = zeros(size(maxarc,2),n);

for i=1:size(maxarc,2)
    index = find(con(maxarc(i,:),:)>=1);
    deletedarc(i,1:size(index,2)) = index;
end
% form an array including all the root nodes covered by C
logical = false;
p = size(deletedarc,1);
c2 = deletedarc(1,:);
for i=1:p-1
    a = c2;
    b = deletedarc(i+1,:);
    c2 = b;
    for x=1:size(a,2)
        for y=1:size(b,2)
            if a(x)==b(y)
                logical = true;
                break
            end
        end
    end
    if logical==0
        c2 = [c2 a(x)];
    else logical = false;
        continue
    end
end
end

```

```

end

c2 = sort(c2);
storedarc = c2(find(c2~=0));

% delete the arcs from presently covered root nodes to the previous marked sensors
for i=1:m
    if nnz(i==C(1:count-size(maxarc,2)))>0
        con(i,storedarc(find(con(i,storedarc)>=1))) = 0;
    end
end

uncov=n-nnz(sum(con_backup(C(2:size(C,2)),:),1)); % uncovered root nodes exist?
if uncov~=0
    % mark all the arcs from presently covered root nodes to all unmarked sensors
    (not in C)
    % and store these arcs in buffer (denoted "2")
    for i=1:m
        if nnz(i==C) == 0
            con(i,storedarc(find(con(i,storedarc)>=1))) = 2;
        end
    end
end

else % no uncovered root nodes
    C_new = [];
    for i=2:size(C,2)
        if nnz(con(C(i,:),:))~=0
            C_new = [C_new C(i)]; % select and unmark the marked C with no arcs
        else
            count=count-1;
        end
    end
end
count=count-1;
break
end
end

```

Unobservability Optimization:

```
% Algorithm for solving unobservability minimization problem
% under single-fault assumption

clc
clear
% Taking number of root nodes from user
n=input('Please enter the number of root nodes = ');

% Taking number of key components from user
m=input('Please enter the number of key components = ');
con=zeros(m,n);
for p=1:m
    for q=1:n
        con(p,q)=input(['Arcs between key components and root nodes
[C',num2str(p),'],'[R',num2str(q),'] = ']);
    end
end

% combine the root nodes that are connected to the same sensor set
column = [];
for i = 1:n-1
    record = [];
    for j = i+1:n
        sumcol = sum(con(:,[i j]),2); % add two columns up
        if nnz(sumcol==1)==0 % identical columns consisting of 0 and 2
            if nnz(i==record)==0 % no i in the record
                record = [record i];
            end
            if nnz(j==record)==0
                record = [record j];
            end
        end
    end
    if isempty(record)~=1 % if record is not empty
        last = record(size(record,2));
        if nnz(last==column)==0
            column = [column last];
        end
    else % if record is empty
        if nnz(i==column)==0
            column = [column i];
        end
    end
    if i==n-1
        if nnz(i+1==column)==0
            column = [column i+1];
        end
    end
end
% array "column" consists of root nodes that have different sensor set.

con = con(:,column); % combine indistinguishable faults
```

```

[m,n] = size(con);
fop = fop(columnn);

% creating set B for all columns of the new connectivity matrix for single
% fault isolation
B = []; fopB = [];
for i=1:n-1
    for j=i+1:n
        sumcol = sum(con(:,[i j]),2);
        sumcol(find(sumcol==2))=0;
        B = [B sumcol];
        fopB = [fopB min(fop(i),fop(j))];
    end
end

con = [con B];
fop = [fop fopB];
[m,n] = size(con);

CS=zeros(m,1);OS=zeros(m,1);
C=0;OC=0;
Ctotal=1500;
stp = sfp.^OS;
unob = fop;
for j=1:n
    for i=1:m
        if con(i,j)==1
            unob(j)=unob(j)*stp(i);
        end
    end
end
systemunob = max(unob);

redun = zeros(m,1);
for i=1:m
    for j=1:n
        if con(i,j)==1
            redun(i)=redun(i)+unob(j);
        end
    end
end

while C<Ctotal

    criticalfault = find(unob==max(unob));
    criticalfault = criticalfault(1);
    Q = find(con(:,criticalfault)==1);
    Cavail = Ctotal-C;
    Q = Q(find(cost(Q)<=Cavail));
    if isempty(Q)==1 % if Q is empty
        break
    else
        Set1 = Q(find(sfp(Q)==min(sfp(Q))));
        Set2 = Set1(find(redun(Set1)==max(redun(Set1))));
    end
end

```

```

Set3 = Set2(find(cost(Set2)==min(cost(Set2))));
CS(Set3(1)) = CS(Set3(1))+1;
C = C+cost(Set3(1));

stp = sfp.^CS;
unob = fop;
for j=1:n
    for i=1:m
        if con(i,j)==1
            unob(j)=unob(j)*stp(i);
        end
    end
end
redun = zeros(m,1);
for i=1:m
    for j=1:n
        if con(i,j)==1
            redun(i)=redun(i)+unob(j);
        end
    end
end

%if (systemunob>max(unob)) && (abs(systemunob-max(unob))>=1e-10)
    OS = CS; OC = C;
    systemunob = max(unob);
%end
end
end
OS
OC
systemunob

```

Appendix G: PCA Fault Detection and Isolation Codes

```
% FDI codes for MSF system
```

```
load MSFerror_2stage_single
load MSFerror_2stage_dual
```

```
% Single-fault directions
```

```
trn = F1error';
[pc_f1,eig_f1,per_f1]=pcacov(cov(trn));
figure;
bar(F1error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;
```

```
trn = F2error';
[pc_f2,eig_f2,per_f2]=pcacov(cov(trn));
figure;
bar(F2error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;
```

```
trn = F3error';
[pc_f3,eig_f3,per_f3]=pcacov(cov(trn));
figure;
bar(F3error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;
```

```
trn = F4error';
[pc_f4,eig_f4,per_f4]=pcacov(cov(trn));
figure;
bar(F4error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;
```

```
trn = F5error';
[pc_f5,eig_f5,per_f5]=pcacov(cov(trn));
figure;
bar(F5error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;
```

```
trn = F6error';
[pc_f6,eig_f6,per_f6]=pcacov(cov(trn));
figure;
bar(F6error);
```

```

xlabel('Sensor');
ylabel('Fault Residuals');
grid;

trn = F7error';
[pc_f7,eig_f7,per_f7]=pcacov(cov(trn));
figure;
bar(F7error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;

trn = F8error';
[pc_f8,eig_f8,per_f8]=pcacov(cov(trn));
figure;
bar(F8error);
xlabel('Sensor');
ylabel('Fault Residuals');
grid;

pc_f_single =
[pc_f1(:,1),pc_f2(:,1),pc_f3(:,1),pc_f4(:,1),pc_f5(:,1),pc_f6(:,1),pc_f7(:,1)
,pc_f8(:,1)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dual-fault directions

trn = F12error';
[pc_f12,eig_f12,per_f12]=pcacov(cov(trn));

trn = F13error';
[pc_f13,eig_f13,per_f13]=pcacov(cov(trn));

trn = F14error';
[pc_f14,eig_f14,per_f14]=pcacov(cov(trn));

trn = F15error';
[pc_f15,eig_f15,per_f15]=pcacov(cov(trn));

trn = F16error';
[pc_f16,eig_f16,per_f16]=pcacov(cov(trn));

trn = F17error';
[pc_f17,eig_f17,per_f17]=pcacov(cov(trn));

trn = F18error';
[pc_f18,eig_f18,per_f18]=pcacov(cov(trn));

trn = F23error';
[pc_f23,eig_f23,per_f23]=pcacov(cov(trn));

trn = F24error';

```

```

[pc_f24,eig_f24,per_f24]=pcacov(cov(trn));

trn = F25error';
[pc_f25,eig_f25,per_f25]=pcacov(cov(trn));

trn = F26error';
[pc_f26,eig_f26,per_f26]=pcacov(cov(trn));

trn = F27error';
[pc_f27,eig_f27,per_f27]=pcacov(cov(trn));

trn = F28error';
[pc_f28,eig_f28,per_f28]=pcacov(cov(trn));

trn = F34error';
[pc_f34,eig_f34,per_f34]=pcacov(cov(trn));

trn = F35error';
[pc_f35,eig_f35,per_f35]=pcacov(cov(trn));

trn = F36error';
[pc_f36,eig_f36,per_f36]=pcacov(cov(trn));

trn = F37error';
[pc_f37,eig_f37,per_f37]=pcacov(cov(trn));

trn = F38error';
[pc_f38,eig_f38,per_f38]=pcacov(cov(trn));

trn = F45error';
[pc_f45,eig_f45,per_f45]=pcacov(cov(trn));

trn = F46error';
[pc_f46,eig_f46,per_f46]=pcacov(cov(trn));

trn = F47error';
[pc_f47,eig_f47,per_f47]=pcacov(cov(trn));

trn = F48error';
[pc_f48,eig_f48,per_f48]=pcacov(cov(trn));

trn = F56error';
[pc_f56,eig_f56,per_f56]=pcacov(cov(trn));

trn = F57error';
[pc_f57,eig_f57,per_f57]=pcacov(cov(trn));

trn = F58error';
[pc_f58,eig_f58,per_f58]=pcacov(cov(trn));

trn = F67error';

```

```

[pc_f67,eig_f67,per_f67]=pcacov(cov(trn));

trn = F68error';
[pc_f68,eig_f68,per_f68]=pcacov(cov(trn));

trn = F78error';
[pc_f78,eig_f78,per_f78]=pcacov(cov(trn));

pc_f_dual =
[pc_f12(:,1),pc_f13(:,1),pc_f14(:,1),pc_f15(:,1),pc_f16(:,1),pc_f17(:,1),pc_f
18(:,1), ...

pc_f23(:,1),pc_f24(:,1),pc_f25(:,1),pc_f26(:,1),pc_f27(:,1),pc_f28(:,1), ...
pc_f34(:,1),pc_f35(:,1),pc_f36(:,1),pc_f37(:,1),pc_f38(:,1), ...
pc_f45(:,1),pc_f46(:,1),pc_f47(:,1),pc_f48(:,1), ...
pc_f56(:,1),pc_f57(:,1),pc_f58(:,1), ...
pc_f67(:,1),pc_f68(:,1),pc_f78(:,1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fdata = error';
m = size(fdata,1);
if max((abs(fdata))) < 0.005
    figure;
    plot(error);
    title(' Residuals');
    disp(['There is no fault']);
else
f_type = ['Inlet cooling brine temperature sensor (F1)      ';
          'Inlet cooling brine flow sensor (F2)             ';
          'Inlet cooling brine salinity sensor (F3)          ';
          'Flashing stage #1 heat transfer degradation (F4)';
          'Flashing stage #2 heat transfer degradation (F5)';
          'Top brine temperature sensor (F6)                ';
          'Top brine temperature controller (F7)            ';
          'Last stage level controller (F8)                 '];
f_device = [];
for j = 1:m
    for i= 1:size(pc_f_single,2)
        ftest = fdata(j,:);
        ff_residual =
ftest*(eye(size(ftest,2))-pc_f_single(:,i)*pc_f_single(:,i)');
        ff1=ff_residual*ff_residual';
        fdist= ftest*ftest';
        f_ind_single(j,i) = 1-sqrt(ff1/fdist);
    end
    ind_f = find(f_ind(j,:)== max(f_ind(j,:)));
    f_device = [f_device;f_type(ind_f,:)];
end
figure;bar(f_ind_single');
xlabel(' directions');
ylabel(' confidence');

for j = 1:m
    for i= 1:size(pc_f_dual,2)

```

```

    ftest = fdata(j,:);
    ff_residual = ftest*(eye(size(ftest,2))-pc_f_dual(:,i)*pc_f_dual(:,i)');
    ff1=ff_residual*ff_residual';
    fdist= ftest*ftest';
    f_ind_dual(j,i) = 1-sqrt(ff1/fdist);
end
ind_f = find( f_ind(j,:)== max(f_ind(j,:)));
f_device = [f_device;f_type(ind_f,:)];
end
figure;bar(f_ind_dual');
xlabel(' directions');
ylabel(' confidence');
end

```

Vita

Fan Li was born in Mianyang, Sichuan Province, People's Republic of China on April 27, 1980. He grew up in Guangyuan, Sichuan, where he attended the primary and secondary schools. He then went to the Tsinghua University in Beijing, where he graduated with a Bachelor of Science degree in July 2002, and Master of Science degree in July 2004, both in Nuclear Engineering. In August of 2005, he entered the Nuclear Engineering program at the University of Tennessee, Knoxville for his graduate studies. He received his Master of Science degree in Nuclear Engineering from the University of Tennessee in December 2007. He will conclude his Ph.D. degree in Nuclear Engineering in May 2011.

He has completed the following publications based on his research work during his graduate studies at UTNE:

■ **Li, F.**, and Upadhyaya, B.R., “*Design of Sensor Placement for an Integral Pressurized Water Reactor Using Fault Diagnostic Observability and Reliability Criteria*,” Nuclear Technology, vol. 173, no. 1, pp. 17-25, 2011.

■ **Li, F.**, and Upadhyaya, B.R., “*Development of an Optimal Sensor Placement Strategy for Nuclear Power Systems*,” Proceedings of the 7th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human Machine Interface Technologies (NPIC&HMIT 2010), Las Vegas, NV, November 7-11, 2010.

■ **Li, F.**, Upadhyaya, B.R., and Coffey, L.A., “*Model-based monitoring and fault diagnosis of fossil power plant process units using Group Method of Data Handling*,” ISA Transactions, vol. 48, no. 2, pp. 213-219, 2009.

■ Upadhyaya, B.R., Perillo, S.R.P., Xu, X., and **Li, F.**, “*Advanced Control Design, Optimal Sensor Placement, and Technology Demonstration for Small and Medium Nuclear Power Reactors*,” Proceedings of the 17th International Conference on Nuclear Engineering (ICONE 17), Brussels, Belgium, July 12-16, 2009.

■ **Li, F.**, Upadhyaya, B.R., and Hines, J.W., “*Optimal Sensor Placement Strategy for the*

Steam Generator System in an Integral Pressurized Water Reactor,” Invited Paper in “Best of NPIC&HMIT” Session, Transactions of the American Nuclear Society, vol. 100, pp. 213-214, June 2009.

■ **Li, F.**, and Upadhyaya, B.R., “*Design of Sensor Placement for the IRIS Reactor Using Fault Diagnostic Observability and Reliability Criteria,*” Proceedings of the 6th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human Machine Interface Technologies (NPIC&HMIT 2009), Knoxville, TN, April 5-9, 2009.

■ Upadhyaya, B.R., **Li, F.**, Coffey, L.A., Kephart, R., and Samardzija, N., “*Development of Data-Driven Modeling Methods for Monitoring Coal Pulverizer Units in Power Plants,*” Proceedings of the 17th Annual Joint ISA POWID/EPRI Controls and Instrumentation Conference and 50th Annual ISA POWID Symposium, Pittsburgh, PA, June 10-15, 2007.