FAULT DETECTION AND DIAGNOSIS OF A NUCLEAR POWER PLANT USING ARTIFICIAL NEURAL NETWORKS

by

Boon Chong Hwang

B.S.E.E., Florida Institute of Technology, June, 1990

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
March 1993

APPROVAL

Name: Boon Chong Hwang

Degree: Master of Applied Science

Title of thesis: **Fault Detection and Diagnosis of a Nuclear Power Plant using Artificial Neural Networks**

Examining Committee: Dr. John Jones, Chairman

Dr. Mehrdad Saif
Senior Supervisor

Dr. Shahram Payandeh
Supervisor

Dr. George Bojadziev
External Examiner

Date Approved: 3/31/1993

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted be me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

"Fault Detection and Diagnosis of a Nuclear Power Plant Using Artificial Neural Networks"

Author:

_____
(signature)

Boon Chong HWANG
(name)

March 31, 1993
(date)

ABSTRACT

Fault detection, isolation and accommodation(FDIA) have always been an important aspect of control system design. Various design techniques such as hardware redundancy, analytical redundancy and expert systems have been used to enhance system performance. Recently, artificial neural networks(ANN) have been highlighted for their potential ability in feature(fault) recognition. Due to their learning capabilities and their inherent parallel structures, ANN are a promising method for fault-tolerant control system design. In this thesis, an approach based on neural networks and mathematical models for detecting and diagnosing instrument failures in nuclear reactors is presented. The reactor's mathematical model is that of H.B. Robinson's nuclear plant located in North Carolina, which produces 2200mw(th) at full power. Multilayer neural networks are used at the first level for identification of plant parameters and at the second level for distinguishing parameter variations from possible faults, and as a pattern recognizer in the third level for the detection of faulty instruments. The design approach was able to simultaneously classify single and multiple anomalies such as sensors and actuators under plant parameter uncertainties. Simulation results presented reveal the promise of artificial neural networks for improving the operating characteristics of nuclear power plants.

DEDICATION

To my family

and

friends.

ACKNOWLEDGMENTS

I would like to express my sincerest appreciation and gratitude to my senior supervisor Dr. Mehrdad Saif for suggesting the research topic and for his advice and encouragement throughout my study. I am very thankful to Drs. John Jones, Shahram Payandeh and George Bojadziev for serving on the committee.

I would also like to thank all my wonderful friends, especially the Shaquills (Jas Parmar, Aziz Rahman, Bob Singh, Jag Mann, Harry Sadhra and Sandy Dhami), the French Connection (Louis Cossette, Jean Varaldi, Yves Audet and Louis Brassard), Chris Pampakerides, Tomomi Matsui and Klover Olsen for all the fun and marvelous time.

Special thanks go to my sisters, Soo Kee and Soo Teng Hwang for their emotional support. Most of all I want to thank my parents, Kim Lian Aow and See Hoo Hwang, for teaching me the importance of doing my best at whatever I do, and for providing me with a supportive environment in which to do it.

Financial support for this research was provided by NSERC and the Center for System Science (CSS) at Simon Fraser University.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

ANN	Artificial neural networks
FDI	Fault detection and identification
FDIA	Fault detection, identification and accommodation
NFDI	Neural networks based FDI
FII	Fault isolation and identification
GDR	Generalized Delta Rule
HU	Hidden unit
PI	Parameter identification
PWR	Pressurized water reactor
PDP	Parallel distributed processing
TLG	Threshold logic generation

NOMENCLATURE

a	Activation value
$f(a)$	Activation function
α	Momentum term to smooth weight changes
η	Learning rate constant
E_p	The mean square error
w_{pij}	Weight from jth node to iith node of pattern p
$\delta_j^{(l)}$	Learning signal
ΔW_{pij}	Adjustment of w_{pij}
y_{pk}	Output of node k when an input pattern p is fed
θ_j	Bias of node j
Δa_{ij}	Deviation of element ij of A
$\Delta\hat{\mathrm{A}}$	Compensated term
δ	Deviation
δP	Deviation of reactor power from the initial steady state value
β	Total delayed neutron fraction
Λ	Neutron generation time
λ_i	Delayed neutron decay constant for ith delayed neutron group
α_p	Coolant pressure coefficient of reactivity
P_0	Initial steady power level
δP_p	Primary system pressure
$\delta\rho_{rod}$	Reactivity due to control rod movement
α_c	Coolant temperature coefficient of reactivity
F_{ci}	Reactivity importance for temperature changes of ith coolant node
α_f	Fuel temperature coefficient of reactivity
F_{fi}	Reactivity important for temperature changes in the ith fuel node
Q_{fi}	Fraction of the total reactor power generated in fuel node i

$(MC_p)_{fi}$	Total heat capacity in the ith fuel node
$(MC_p)_{ci}$	Total heat capacity of both coolant nodes associated with the ith fuel node
A_f	Heat transfer area
τ	Residence time
P_p	Pressure in the pressurizer
δW_w	Mass flow of the water
δq	Change in the rate of heat addition to the pressurizer fluid
τ_{SG}	Residence time of coolant node in the steam generator
h_{pm}	Heat transfer coefficient for primary coolant to metal
h_{ms}	Heat transfer coefficient for metal to secondary coolant
M_m	Mass of tube metal
C_m	Specific heat of tube metal
D_i, B_i	Coefficients
P	Reactor power
C	Precursor concentration
T_f	Fuel temperature
T_{ci}	Coolant temperature in ith node
T_p	Temperature of primary coolant node in steam generator
T_m	Tube metal temperature
P_s	Steam pressure
T_{up}	Reactor upper plenum temperature
T_{HL}	Hot leg temperature
T_{Ip}	Primary coolant temperature in steam generator inlet plenum
T_{op}	Temperature of primary coolant in steam generator outlet plenum
T_{CL}	Cold leg temperature
T_{Lp}	Reactor lower plenum temperature
W_{so}	Steam flow rate to turbine
$r(t)$	Output residual used for PI network
$\Delta y(t)$	Output residual used for FII network
$()_n$	Nominal value
$\hat{()}$	Compensated value
$\Delta()$	Residual
$\Delta\tilde{()}$	New residual obtained from feature extraction
Γ_i	Influent constant of the ith sensor

CHAPTER 1

INTRODUCTION

Many advanced process plants and machines are extremely complex, and always depend on automatic control for satisfactory operation. In order to achieve and maintain system stability and assure satisfactory and safe operation, there is increasing demand for dynamic systems to continue acceptable operation following failures. Therefore, failures detection, identification, and accommodation (FDIA) has always been an important aspect of fault tolerant control system design. In a nuclear power plant, for example, tens of alarms can occur in a few second after a fault. Locating the fault might be of utmost importance due to safety, political and other reasons.

1.1 Literature Review

A failure can be described as a variety of malfunctions in the actual plant dynamics that affects the capability of the system to perform its specified mission. When a

failure occurs, it must be detected as early as possible. The detection task is the process of indicating the presence of a failure. After a failure is detected, the faulty component must be isolated from the system. The identification task then estimates the magnitude of the failure. In addition, we can define failure diagnosis as the task of isolating and identifying the failures. After diagnosing, the failure should be accommodated by system reconfiguration.

Traditionally, physical or hardware redundancy is used to accomplish fault-tolerance in dynamic systems. In order to provide protection against malfunctions in control systems, one of the easiest alternatives is to install redundant hardware which perform the same task. Majority vote ruling scheme is then used to detect and isolate failures. The main drawbacks encountered in implementing hardware redundancy are increased cost and maintenance. Furthermore, such a system requires the additional space to accommodate instruments.

Although physical redundancy are capable of protecting against control system instrument failures (sensors and actuators), it does not generally address failures of plant components. Due to the availability of reliable and powerful computers, analytical redundancy has been developed which has the advantage of requiring fewer additional components. The principal idea of analytical redundancy is that the system's mathematical model, along with the input and output measurements can be used to generate quantities termed residuals. If the system model is perfectly consistent, the residuals will be near zero under normal operating conditions, otherwise the residuals will diverge at the presence of an anomaly. Analytical redundancy techniques have been discussed in survey papers by Willsky[52], Isermann[22], Basseville[3], Frank[10], Merrill[34] and Stengel[46].

A variety of FDIA techniques using analytical redundancy can be found in literature. Beard[4] and Jones[23] used a detection filter for fault detection. The detection filter was designed so that the magnitude and the direction of its error vector in the output space would provide information regarding the possibility of sensor failures, as well as the faulty sensor, respectively. Mehra and Peshon[33] proposed an innovation based test using a single Kalman filter. Their approach indicates the presence of a faulty sensor only, it can not isolate the faulty sensor. Deckert et al.[8] presented the parity space approach which was implemented for sensor failure detection aboard the F-8 Digital Fly By Wire (DFBW) aircraft. Clark[6,7] presented the dedicated observer scheme (DOS) where Luenberger observers were dedicated to individual sensors. Since the introduction of DOS, other approaches based on it have been proposed (Frank[10], Saif and Villaseca[44,45]).

In general, most plant models used for dynamic system design are uncertain. Even if an exact model is available, it may be so complicated that it must be approximated by a simple, but uncertain, design model. For example, non-linear models may be linearized for small deviations from an operating condition. Furthermore, physical parameters of the plant and its environment may be uncertain.

In an uncertain model system, the effect of modelling errors obscures the effect of faults generating a source of false alarms. Hence, robust FDIA techniques should be explored[10].

Various effective techniques to FDIA using analytical redundancy which are insensitive to modelling errors have been developed. Frank and Keller[11] used two dedicated observers to distinguish parameter variations and instrument malfunctions. However, this approach is restricted to single input single output (SISO) systems.

Watanabe and Himmelblau[50] applied the extended Kalman filter to identify process parameters indicative of process faults caused by the deterioration of components. The major difficulty with FDIA based on the earlier techniques is that the approach involves a massive computational load for estimation; furthermore, multiple failures are unable to be detected and recognized simultaneously.

Another alternative to FDIA under uncertainties in the system's model is based on the theory of unknown input observer (UIO). Unknown input observers are a class of estimators designed to provide an accurate estimation of the system's state in the face of unknown exogenous plant disturbances in the system's dynamical model. The theory of UIO was first examined by Basile and Marro[2]. Later, Kurek[31] gave necessary and sufficient conditions for the existence of UIO. The design of UIO generally requires geometric approaches or a solution of complicated matrix equations. Recently, Guan and Saif[12,13,43] provided a very straight forward, but yet elegant approach to the design of UIOs which was used to detect and identify the faults of a VTOL Helicopter in the vertical plane.

The major drawback in the context of robust FDIA using UIO is that the existence condition for the UIO must be satisfied. If an UIO does not exist, it is impossible to detect and identify the instrument failures.

In recent years, use of knowledge-based expert systems in fault-tolerant control system design has increased significantly. By relying upon device models and device interconnections, a knowledge-based expert system can locate a disturbance origin (e.g., a device malfunction) by searching for a casual path along with the disturbance propagates its effects. Various FDIA techniques via artificial intelligence have been

developed. For example, Palovitch and Kramer[30] have proposed a fault diagnostic expert system for chemical plants based on cause-and-effect models. Kim and Modarres[26] used a Goal Tree-Success Tree (GTST) model to represent a deep and hierarchical human cognitive knowledge. Handelman and Stengel[14] have included quantitative and qualitative reasonings in their rule-based failure diagnostic system. Later Huang and Stengel[21] employed structural and behavior descriptions to establish discrepancy, detection and causal interaction in diagnosing the flight control system of a CH47 tandem-rotor helicopter.

FDIA using knowledge-based expert systems have some drawbacks associated with them. They lack plant generality and they tend to fail under novel circumstances[38]. The rule-based knowledge bases are also difficult to understand as they lack higher-level representation and organization of plant knowledge. As a result, they involve substantial effort in development which can be quite time-consuming. Furthermore, they are not able to learn and improve system performance dynamically.

Artificial neural networks have gained enormous popularity in the last few years. They have been applied to a variety of domains including control, monitoring and diagnostics. Trained networks offer considerable promise as fast and robust computational models that implicitly take into account the nonlinear behavior of a system and can be used in a predictive manner for model-based fault-tolerant control. Recently, artificial neural networks have been used to represent knowledge about failures and do the pattern recognition tasks. First, a variety of fault situation are used to train the neural net. After the network has learned them, it can do the proper fault detection and diagnosis. Hoskin and Himmelblau[20] developed an artificial neural network based diagnostic system for a chemical process consisted of a series of three

continous-stirred tank reactors. Watanabe et al.[51] used an artificial neural network approach to estimate the degree of failures in a chemical reactor. Venkatasubramanian and Chan[49] presented a binary-input network to diagnose faults of a fluidized catalytic cracking process. Later Naidu et al.[35] studied a sensor failure detection system based on a multilayer perceptron network and backpropagation learning algorithm. Recently, Upadhaya et al.[9,47,48] utilized a backpropogation network to develop "models" of signals from both a commercial power plant and the EBR-II.

Most of the aforementioned works are applied to very slow chemical process and the faults are detected only when the system is in steady state. In this thesis, an artificial neural network based FDIA technique for dynamical systems is presented. The approach is based on the fact that the model of the dynamical system under plant parameter uncertainties and instrument failures can be used to teach neural networks for parameter identification, and fault detection and diagnosis.

1.2 Thesis Overview

This thesis is divided into six chapters and one appendix. Chapter Two, entitled "FDIA overview and background," addresses the general ideas of a fault-tolerant control system. Chapter Three, entitled "Artificial neural networks," addresses the unifying principle, mechanisms and structures of neural networks. In chapter Four the neural networks based scheme for parameter identification and fault detection and diagnosis are presented. In chapter Five simulation studies are performed on the model of a pressurized water reactor (PWR). Finally, chapter Six, entitled "Summary and conclusions," investigates the effectiveness of the proposed artificial neural network

based FDI technique in enhancing simulated fault tolerance. Some of the advantages resulting from the use of ANN based FDI techniques are reviewed, as well as the limitations. In a final word, suggestions for future research work are presented. The appendix provides additional details not found in the main text.

CHAPTER 2

FDIA OVERVIEW AND BACKGROUND

2.1 FDIA Problem Characteristics

The important requirements for reliable automated fault detection and diagnosis in modern technological installations are increasing rapidly as the systems become more complex and they need to operate with minimum malfunctioning or breakdown time. For example, in a chemical industrial plant, product quality is maintained by assuring that process variables fluctuate within permissible ranges. If operating conditions go outside these ranges, the product quality is not acceptable, or more critically some catastrophic event might result. The early detection of plant's failure could prevent system malfunction or serious damage, which could also lead to disaster. Consequently, failure detection, identification and reconfiguration plays a very important

role in the system design of complex plants such as space vehicles, aircrafts, submarines and nuclear power plants, where safety is always the highest priority. The following overview addresses the various steps that are required for tackling FDIA problem.

2.2 FDIA Schemes

The schemes used in fault detection, identification and accommodation are usually divided into two general categories:

2.2.1 Mathematical Model-Free Approaches

No mathematical model of the process is necessarily needed in these approaches for fault detection and diagnosis. Physical redundancy and limit checking are the typical examples of using these methods. However, these approaches require extra expenses and additional space.

Recently, knowledge-based (expert systems) technology has been applied in these methods. An expert system has been defined as a computing system which embodies organised knowledge concerning some specific area of human expertise, sufficient to perform as a skillful and cost effective consultant. This knowledge-based model-free technique acquires the knowledge from experts directly, usually in the form of (if-then) production rules. This type of approach utilizes empirical associations between evidence and conclusions, but does not depend on a "deep" (functional) understanding of the domain itself. This has been referred to as the "shallow" knowledge approach.

This approach is typical in medical diagnosis programs, where underlying disease mechanisms are unknown or difficult to describe (for example, MYCIN is based on this approach). For many reasons, the shallow approach is not ideal for complex plant modelling in the form of expert systems. For example, the knowledge needed to solve a plant upset condition is broad and ill-defined. The required knowledge may include the plant layout, physical and chemical properties of the fluids, design specifications, knowledge of past and current operating conditions, and interpretation of process measurements.

In general, mathematical model-free approaches are unable to indicate incipent failures, and identify the magnitude of the failures.

2.2.2 Mathematical Model-Based Approaches

There are a variety of fault detection and diagnosis methods using a mathematical model of the process. Basically, model based failure detection methods rely on the determination of changes occuring in the system due to the presence of a failure, in comparison to the normal condition of the system. As a result, the concept of analytical redundancy is widely applied in these methods. The estimation techniques of Kalman filtering and the Luenburger observer are usually utilized to generate redundant information. In its simplest form the detection logic would consists of comparison of a sensor measurement with its estimated value for fault detection. This is in contrast to hardware redundancy, where similar parallel measurements from each sensor are compared.

2.3 Model-Based FDIA Procedures

Based on the mathematical model, the schematic conceptual architecture of the FDIA is illustrated in Figure 2.1. There are three neccesary procedures required to achieve the task of FDIA:

2.3.1 Generation of Redundancy Relations

Generally, redundant information regarding the plant is generated by estimation schemes. Failures of components may take the form of abrupt additive in measurement bias. These changes are determined by comparing the estimated variables obtained from the model of the normal process with the measurements of the actual plant dynamics. The difference between these two values are called residuals. If the actual and modelled measurements generally agree, the residuals will be zero or approximate to zero; otherwise, it will be non-zero, and this would indicate the presence of a fault.

2.3.2 Decision Making

After the residuals are generated, they then proceed to a fault decision process with the goal of detecting and diagnosing the anomalies due to the existence of abnormal behavior. This fault decision process is divided into the following three tasks:

1. *Fault detection*, i.e., the indication of the occurence of a failure.

2. *Fault isolation*, i.e., the localisation of components that have failed.

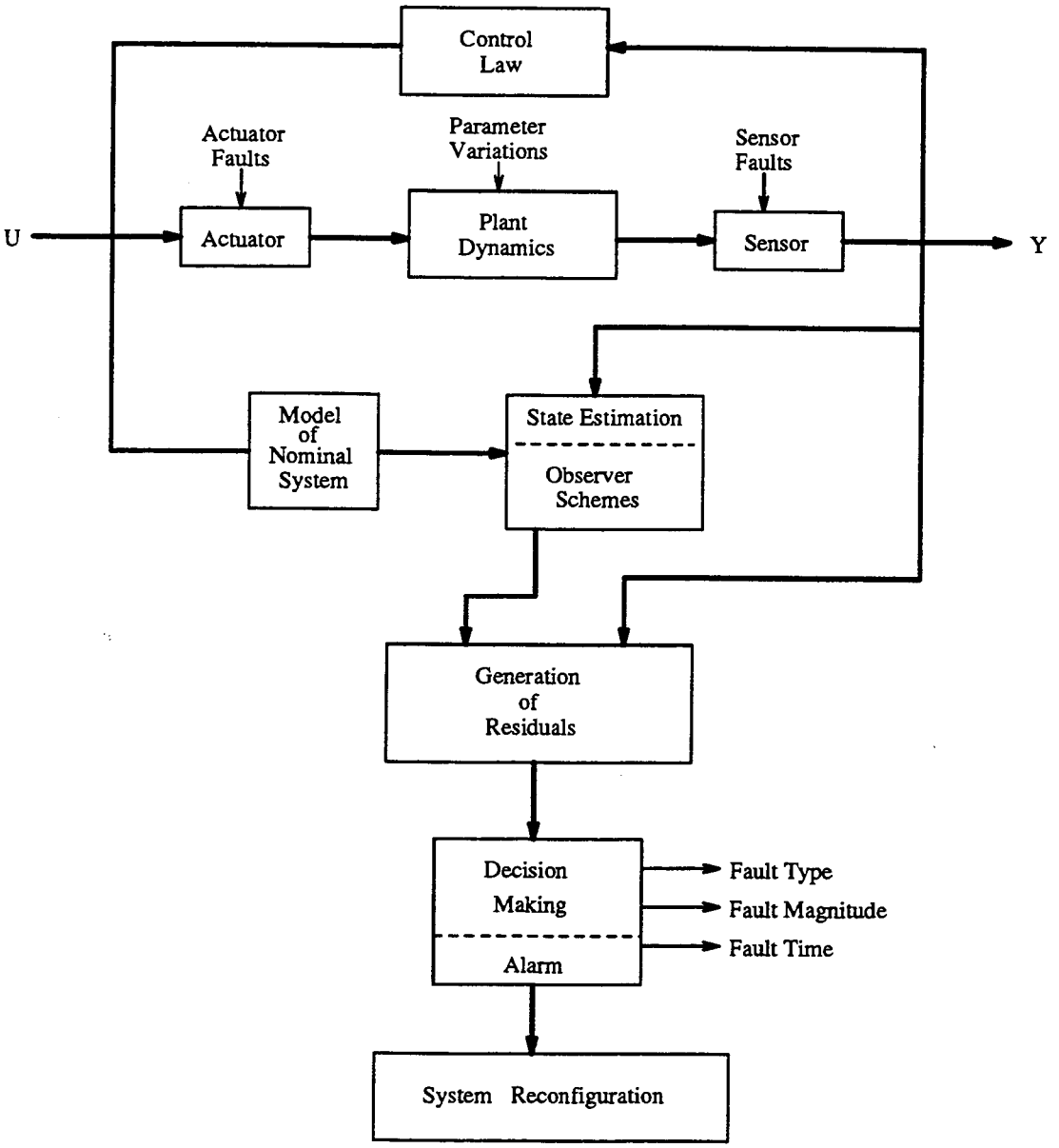


Figure 2.1: General architecture of FDIA

3. *Fault identification*, i.e., the determination of the magnitude and/or the shape of the fault.

2.3.3 System Reconfiguration

After fault decision process, the system must attempt to accommodate the failure(s) and to retain stability and performance characteristics. Failure accommodation could include hardware tasks (e.g., activating back-up systems) and software actions (e.g., adjusting the feedback control gains). However, failure accommodation is not dealt with in this thesis.

2.4 Classification of Failures

A failure can be described as a variety of malfunction in the actual plant dynamics that affects the capabilty of system to perform its specified mission. In general, these malfunctions can be caused by the following three sources:

1. *Insturement failures*:

   These failures usually occur in the sensors and actuators of the plant. Due to their existence, there are discrepancies between the actual and measured values of the system's input or output variables. For instance, in aerospace applications the failure of control actuators may manifest themselves as shifts in the parameters of control gain matrix. Failures of sensors may take the form of abrupt changes in the parameters of the output matrix, or development of additive biases in the measurement.

2. *Disturbances*:

   Disturbances involve the unmeasurable noise contamination on inputs and/or outputs of the plant. These noises are usually random functions originating in the environment, such as a wind gust acting on an aircraft.

3. *Plant parameter uncertainties*:

   The uncertainties usually arise from modelling errors, inexact knowledge of the plant parameters, and parameter variation due to component aging, etc. Most plant models used for system design are uncertain. Even if an exact model is available, it may be so complicated that it must be approximated by a simpler, but uncertain, design model. For example, non-linear models may be linearized for small deviations from an operating condition. However, outside of this range the nonlinearities of the plant produce signals which are not modelled accurately. If not accounted for, such uncertainties and nonlinearities may be interpreted as faults by the fault detection logic.

2.5 Dedicated Observer Schemes

There are a number of different analytical redundancy approaches for FDIA that have been proposed in the literature. The basic idea of this approach is to reconstruct the outputs of the system from the measurements, with the utilization of observers or Kalman filters using estimation error. Among various estimator schemes, the dedicated observer scheme (DOS) proposed by Clark[6,7] is one of the most well known approach for FDIA system.

In the DOS, each sensor is dedicated to one observer or measurement estimator.

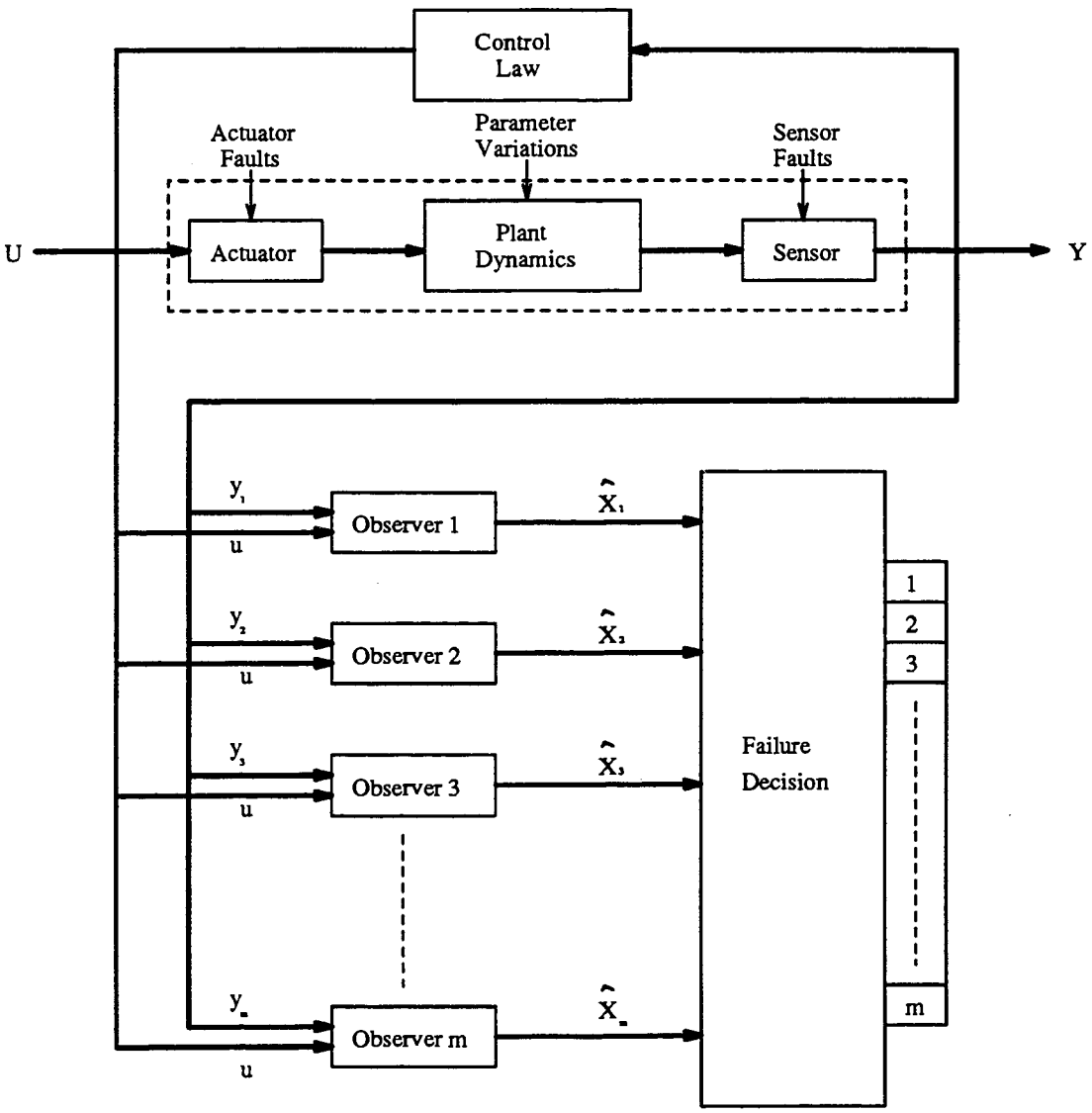


Figure 2.2: Dedicated Observer Scheme

It is assumed that the system is observable from each measured variable of the process. Each dedicated observer estimates the state variables based on the individual measurements and the corresponding input vector **u**. Under no-fault condition, the estimated state variables should converge to the actual state variables after a short transient period. If, for example, a fault occurs in one of the sensors, then the estimate of the related measured variable driven by the corresponding dedicated observer will be erroneous which can be identified by the threshold logic. The structure of dedicated observer scheme is illustrated in Figure 2.2.

The major drawback of the DOS approach is the demand for multiple observers and the correspondingly large computational burden.

2.6 Unknown Input Observer Scheme

A variety of robust FDIA techniques using Unknown Input Observer (UIO) scheme have also been developed in the literature. This technique basically relies on the theory of state estimation for linear systems subject to exogenous unknown inputs and the unknown input observer theory. An unknown input observer is a class of estimators capable of estimating the state of a linear system with unknown inputs. Among various UIO schemes, the design algorithm proposed by Saif and Guan[12,13,43] is believed to be computationally simpler and more direct. Their technique provided a very straight forward, but yet elegant approach to the design of UIOs. In the following, a brief overview of the FDI technique developed by Saif and Guan is given.

In the proposed scheme, only a single UIO is required for the purpose of failure detection and identification . The architecture of the unknown input observer scheme

is illustrated in Figure 2.3. Consider a linear time-invariant system with both instrument failures and plant uncertainties. This system can be represented in the following state space formulation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}\mathbf{v} \tag{2.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{E}\mathbf{f} \tag{2.2}$$

where $x \in \Re^n$ is the state vector, $u \in \Re^m$ is the input vector, $v \in \Re^p$ is the unmeasurable input disturbance which can be considered as actuator failure, and $y \in \Re^q$ is the measurable output vector, and $f \in \Re^r$ is the unmeasurable additive disturbance which can be considered as sensor faults. The system is transformed into a completely known system with instrument faults, and some additive unknown input disturbances. Then the system has the following state space representation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{D}^*\mathbf{v}^* \tag{2.3}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{E}\mathbf{f} \tag{2.4}$$

where

$$\mathbf{D}^* = [\mathbf{D} \quad \mathbf{I}_A \quad \mathbf{I}_B] \tag{2.5}$$

$$\mathbf{v}^* = \begin{bmatrix} \mathbf{v} \\ \Delta\mathbf{A}_a\mathbf{x} \\ \Delta\mathbf{B}_b\mathbf{u} \end{bmatrix} \tag{2.6}$$

where ΔA and ΔB are the plant uncertainty vectors. D^*V^* is the additional part of the system that accounts for uncertainties. This additional part is obtained by lumping all uncertain part of the system matrices together and treating them as additional unknown input to the system. The model of the system with both actuator and sensor failures is then augmented into a higher dimensional model as follows:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{f}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \underline{0} \\ \underline{0} & \mathbf{A_f} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{f} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \underline{0} \end{bmatrix} u + \begin{bmatrix} \mathbf{D^*} & \underline{0} \\ \underline{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v^*} \\ \xi \end{bmatrix} \tag{2.7}$$

$$\mathbf{y} = [\mathbf{C} \quad \mathbf{E}] \begin{bmatrix} \mathbf{x} \\ \mathbf{f} \end{bmatrix} \tag{2.8}$$

where A_f is a rxr matrix and ξ is an unknown input vector. At this point the obtained model is suitable for employment of the UIO. Once the estimate of the augmented state is obtained, the sensor failures can be identified immediately by checking the components of the estimated vector f. Any non-zero component of this vector would indicate the presence of a fault in the corresponding sensor. Moreover, the non-zero value gives the estimate of the fault magnitude. Hence, faulty sensors can be detected, as well, the shape of the fault can also be identified. Since both $\hat{x}$ and $\hat{f}$ are available through estimation, the only unknown element is the unknown input vector v. By solving the equation of (2.3) for v in terms of x and u, the faulty actuator can be easily detected from the non-zero component in v.

The major shortcoming of this FDIA approach is that the existence condition for the UIO must be satisfied. The existence condition requires the number of unknown inputs be less than or equal to the number of outputs. For systems with large amounts of structural uncertainty and not enough available output measurements, it would be impossible to satisfy the existence condition for the UIO, and in such situations the

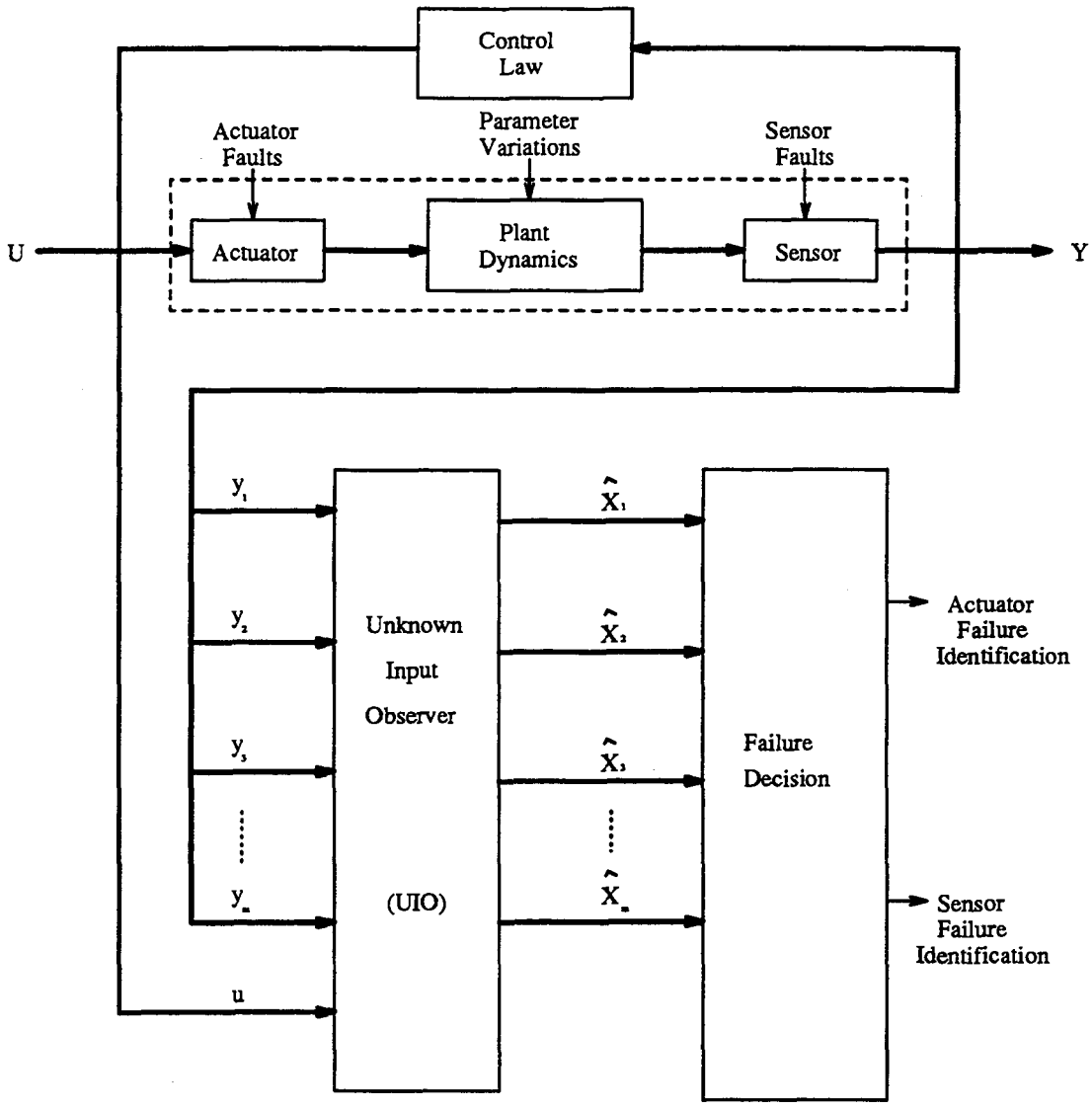


Figure 2.3: Unknown Input Observer Scheme

proposed approach will fail. If an UIO does not exist, it is impossible to detect and identify the instrument failures.

2.7 Analytical Knowledge-Based Methods

Recently, knowledge-based methods (expert systems) combined with the existing analytical and algorithmical FDIA methods have opened a new dimension of possible fault diagnosis for complex processes with incomplete process knowledge. Whilst the algorithmic approaches implement *quantitative* analytical models, the knowledge-based method utilizes *qualitative* models based on the available knowledge of the system. The combination of both strategies is believed to be able to evaluate all available information and knowledge of the system for fault detection and diagnosis.

The general architecture of a model- and knowledge-based FDIA system is shown in Figure 2.3. This FDIA design contains an on-line expert system which combines the analytical model-based technique with the knowledge-based technique using heuristic reasoning. The overall FDIA system consists of the following architectural elements:

1. *The data base*, i.e., information about the present known or deduced facts (states) of the process.

2. *The knowledge base*, i.e., a description of the objects in the knowledge domain and casual and heuristic relationships in the domain. The process knowledge is usually represented by rules and/or frames.

3. *The inference engine*, i.e., the specification of the steps to be taken when applying knowledge to the present state. Generally, a data-driven search (forward

chaining) or a goal-driven search (backward chaining) are used to perform the reasoning tasks.

4. *The explanation component*, i.e., this component provides the user with useful information on why and how the conclusions were obtained.

In this analytical knowledge-based approach, the on-line expert system perform its FDIA tasks by evaluating the analytical and heuristic knowledge of the dynamic plant. These tasks are accomplished in the inference engine which combines quantitative reasoning (algorithmic operations based on analytical redundancy) with qualitative (heuristic) reasoning in a single problem solver. The inference engine executes a form of search common to:

- The analytical knowledge in terms of the mathematical model (structure and parameter).
- Heuristic knowledge of fault propagation, fault statistics, and process operational and environmental condition, etc.
- The actual data (input, output, operating condition, etc.)

A shortcoming of this approach is that the development of the analytical and heuristic knowledge base can be very difficult, expensive and time-consuming. Furthermore, the accurancy of fault detection and diagnosis depends on how rich and precise the knowledge base is.

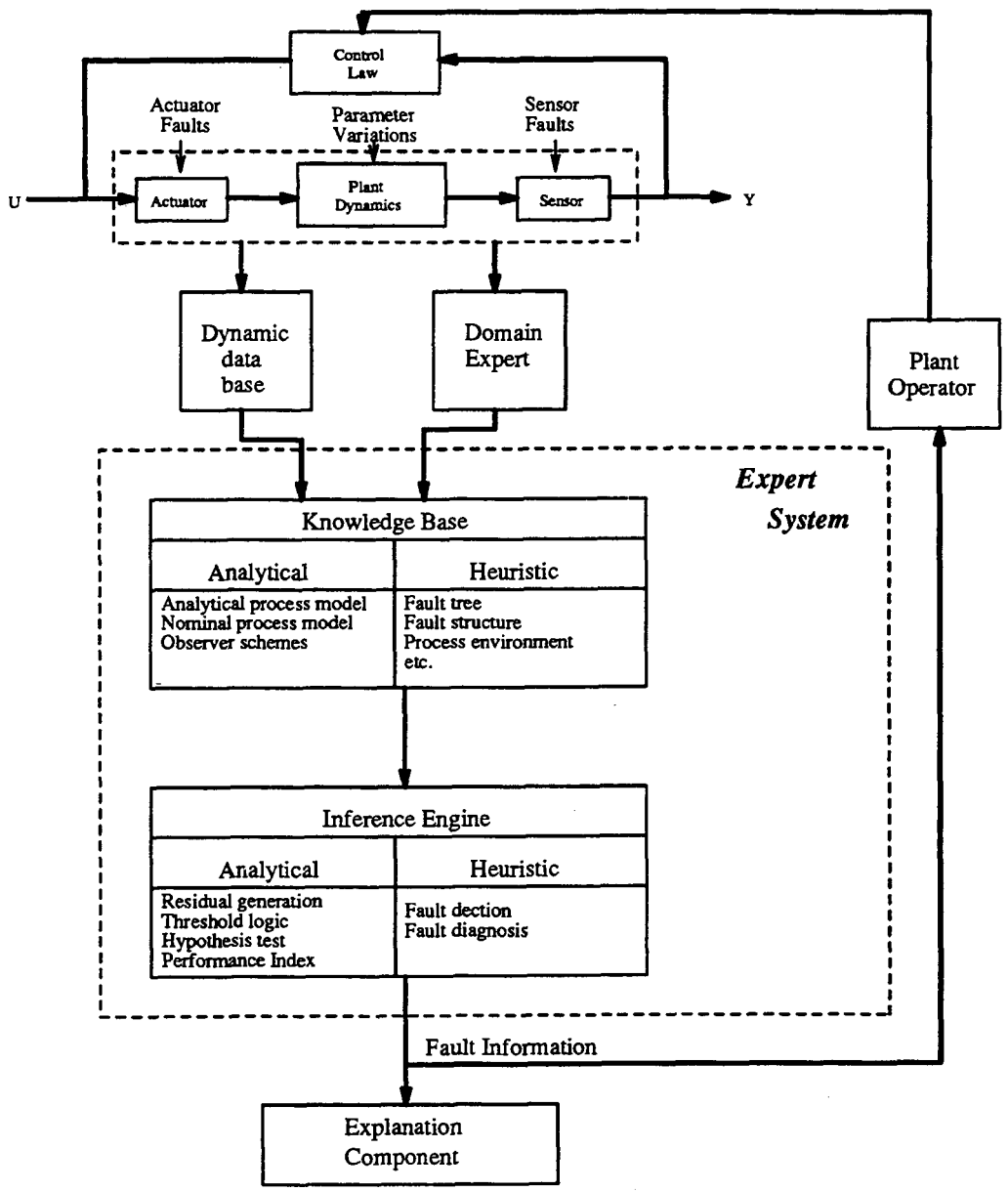


Figure 2.4: Architecture of a model-and knowledge-based FDIA system

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

An artificial neural network (ANN) is a technology which has been inspired by studies of brain and nervous system. A neural network is a collection of simple neuron-like processing elements that process information by their dynamic behavior to external sources. In this chapter, the principles, mechanisms, and architectures of artificial neural network are described.

3.1 Neuron Models

3.1.1 Biological Neuron

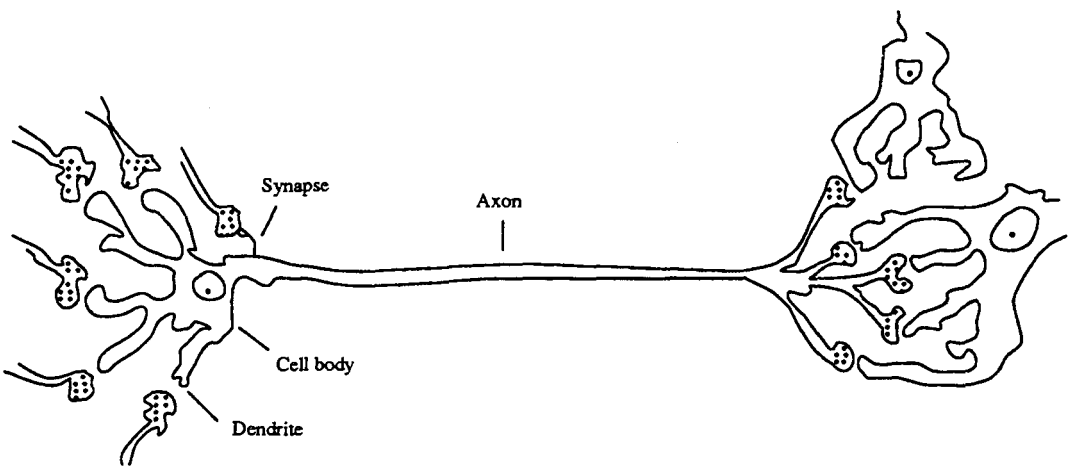


Figure 3.1: Biological Neuron

A biological neuron shown in Figure 3.1 basically consists of four parts: a cell body, dendrites, axons and synapses. The input signals are transmitted through the input fibers called *dendrites*. The cell body, which is called *soma*, processes the signals from the dendrite to produce an output pulse that carries to an output fiber named the axon. The axon connects to the dendrite of other neurons at the synapse. Moreover, the synapse is a specialization of the neuron's membrane which allows the output pulse from the axon to induce a voltage over the receiving dendrite's cell membrane.

3.1.2 Artificial Neuron

In artificial neural networks, artificial neurons (nodes) are very simple processing elements motivated by the neural structure observed in real biological organisms. We can

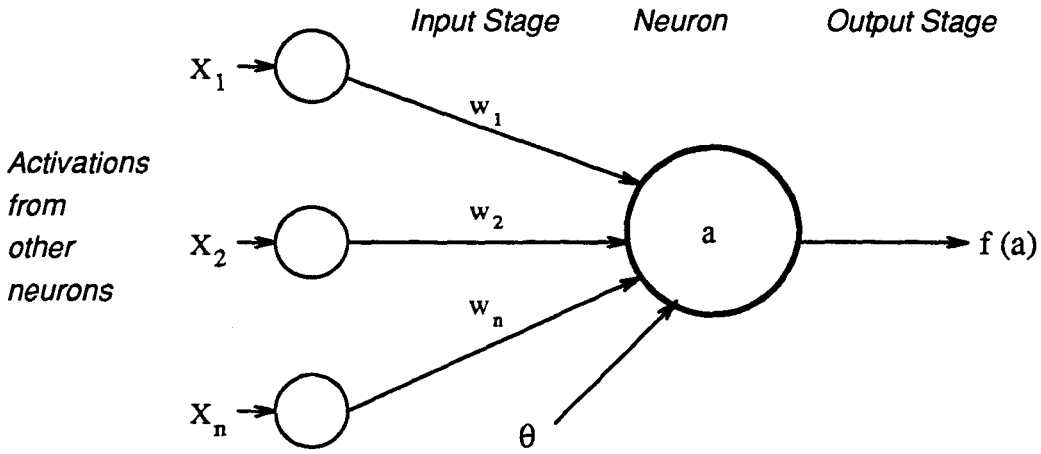


Figure 3.2: Artificial Neuron

describe them as a simple operator performing a mapping from a multidimensional input obtained from other processing nodes or external stimuli to a single dimensional output which is sent to other nodes through weighted connections. Due to the inherently parallel architecture of the neural network, the computations of many nodes can be accomplished simultaneously.

A typical artificial neuron is illustrated in Figure 3.2. The input stage of neurons can be described as the connections that forward to the neuron from other neurons in the network. Each connection has an associated weight with it, which represents the strength of the connection. Each neuron is also associated with an activation value **a**, a number corresponding to its state. Each neuron is also connected to a bias which provides a threshold for the activation of the neuron. The activation was simply a linear weighted sum of the inputs, plus a bias

$$a = \sum_{i=1}^{n} w_i x_i + \theta \tag{3.1}$$

where variable x_i is the incoming input and variable w_i is the weight or the connection strength; and θ is the bias of the neuron. The measurements in the input x_i and the

weights in the connection strength w_i are dimensionless real numbers. The activation values are unbounded real-valued and dimensionless.

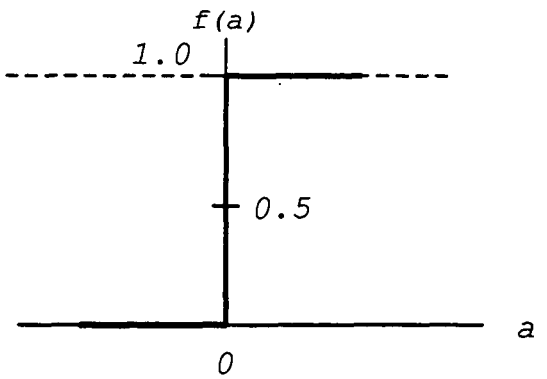
The output stage of the neuron consist of a function, called activation function or output function, $f(a)$. The output function utilized, is a threshold function for which no output is produced unless the neuron's activation reachs a certain threshold value. This threshold quality creates the basis for discriminator functions, which separate the feature space R^m into different decision regions (m is the dimensional space of the input patterns). There are three most commonly used discriminator functions: binary step, linear ramp and sigmoid function, as shown in Figure 3.3.

In this work, a sigmoid logistic function is chosen to calculate the activation of neuron. This activation function is given by

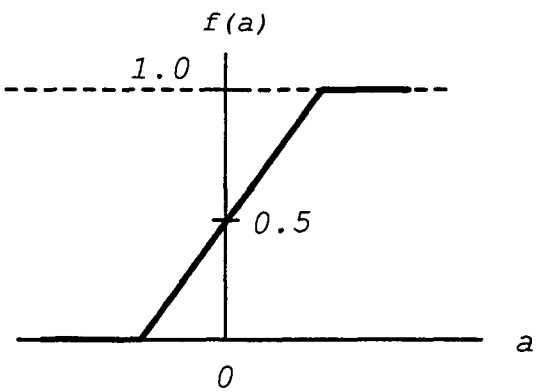
$$f(a) = \frac{1}{1 + e^{-a}} \tag{3.2}$$

where $f(a)$ is the neuron's activation, and **a** is its input. The form of this function is illustrated in Figure 3.3.

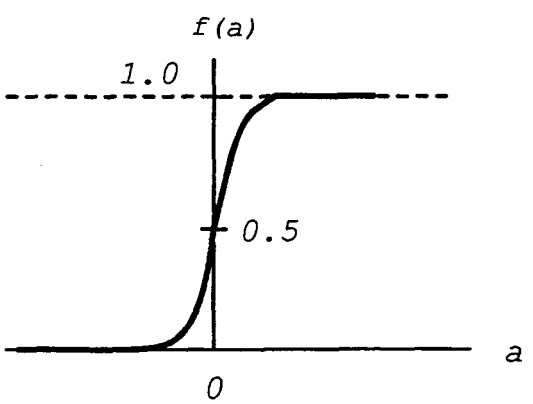
The main reason for selecting eqn 3.2 is that it has been successfully implemented in many pattern recognition problems. Furthermore, since it is differentiable and non-decreasing it can be used with the backpropagation algorithms to train the multilayer networks .



Binary Threshold



Linear Ramp



Sigmoid

Figure 3.3: Commonly Used Activation Functions

3.2 Overview

3.2.1 Type of Neural Networks

The neural networks study began after McCulloch and Pitts[32] introduced the abstract neuron model for performing a simple task in 1943. The major neural networks are listed in Table 3.1. In 1950, F. Rosenblatt[39] presented the perceptron models, which ignited a great amount of research interest in neurocomputing. The perceptron is a single layer feedforward network for pattern classifiers. The inputs were transferred to the neural layer with randomly weighted connections. The neural network performed successfully with application of the Hebb learning rule. The major limitation, pointed out by Minsky and Papert, is that the perceptron cannot classify complex nonlinear categories. Multilayer perceptrons were then developed by Rosenblatt in the 1960's to overcome many of the limitations of single-layer perceptrons, but were generally not utilized in the past, due to the fact that effective training algorithms were not available.

Another major category in neural networks is associative memory. J. Anderson developed "Brain-State-in-Box-Model" with his linear associator and Hebb learning rule[1]. The network consists of a layer with feedback and one postprocessing output layer. Because of the positive feedback architecture and the learning rule, the output is the best-matched pattern from the stored memory for a given input.

In 1982, Hopfield[18] proposed the Hopfield network, which is basically single layered with feedback. The condition for the synapse weighting is very restricted (being symmetric and having no self-feedback terms), while that for the neuron transfer

Name	*Years Introduction*	*Primary Applications*
Perceptron	1957	Typed-character recognition, Simple pattern classifier
Brain-State-in-a-Box (Linear Associator)	1977	Associative memory
Hopfield Network	1982	Image processing, Assoc. memory, Problem solver
Multi-layer with Back-propagation	1974-1985	Wide ranges: Speech synthesizer Pattern recognition
Boltzmann Machine	1985	Pattern recognition for radar, sonar
Bidirectional Associative Memory	1986	Associative memory

Table 3.1: Major neural network models

function is very relax (only monotonically increasing and bounded). Using an energy or Lyapunov type function, Hopfield proved that the network always moves toward a low energy level and hence is stable. The network has been applied to associative memory and many engineering optimization problems.

The multilayer feedforward neural networks are vitalized by the back propagation learning rule. The usefulness of the multilayer neural networks was recognized in the past, but the decision of synapse weightings was the main problem. The multilayer neural network can be used for various applications including data encoding/decoding, data compression, signal processing, pattern classification, and robotic controls.

A Boltzmann machine has a similar network architecture to the Hopfield network, but differs in stochastic update and learning properties. The stochastic update in retrieving and learning processes is based on the simulated annealing technique using the Boltzmann probability function. By decreasing the temperature of the probability function from a high value, the network always finds the global minimum in the energy

surface.

The bidirectional associative memory (BAM) is a generalized Hopfield model to heteroassociate network[29]. BAM has two fully connected central layers and input/output buffer layers. The synapses and neurons in the central two layers are bidirectional. For a given input, the BAM layers oscillate until a stable state is reached. The final stable output is the closest association stored in BAM.

3.2.2 Type of Learning Algorithms

Learning is the process of adapting the synapse weightings in response to external stimuli. Various types of learning algorithms are listed in Table 3.2. The learning rules were developed with the network architectures as shown in Figure 3.4 . The first learning rule, named *Hebb learning rule*, which shows that the neural network can learn for a certain function, was presented in 1957[15]. The rule requires that if an input and output are activated at the same time, the weighting between the input and output is increased. T. Kohonen presented *competitive learning*, where each neuron completes with others at a given input and the winner adapts to get more strength[28]. This type of learning, called *unsupervised learning*, does not need reference data. On the other hand, desired outputs can be given in the supervised learning approach. The *simple delta rule* is applied to adjust the synapse weightings, using the error between the desired output and actual network output. Many derivatives from the simple delta rule are used to efficient learning results. The famous application of supervised learning is backpropagation for a multilayer network. The root-mean-square error at the output layer propagates backward through the network and is

Model	*Basic Equation*	*Comments*
Hebb's Rule	$\Delta w_{ij}=\alpha\ a_i \cdot a_j$	Unsupervised, Hopfield networks
Competitive Learning	$\Delta w_{ij}=\alpha\ a_i \cdot (a_j - w_{ij})$	Unsupervised, Associative memory
Delta Rule	$\Delta w_{ij}=\alpha\ a_i \cdot E_j$	Supervised, Back-propagation
Boltzmann Machine	Boltzmann Probability function with simulated annealing	Supervised, Boltzmann Machine

Table 3.2: Basic learning algorithms

used to update the synapse weighting between layers. The process continues until the input layer is reached. The delta rule or its derivatives are used for determine the synapse weighting modifications. Due to an extremely long learning time, only one or two hidden layers are often used. The counterpropagation network[16], which can be applied only to three-layer networks, combines the competitive learning and delta rule. Thus, the counter-propagation network selects the nearest stored pattern due to the competitive operation inside the hidden layer.

Another type of learning that falls between unsupervised learning and supervised learning is reinforcement learning[27]. In this learning, an external observer gives a response as to whether the network output is good or not. The learning rule of a Boltzmann machine[17] is based on the stochastic process, which constructs representations of the reference patterns with the simulated annealing technique. Due to the cooling process, the retrieving and learning time of the Boltzmann machine is extremely long.

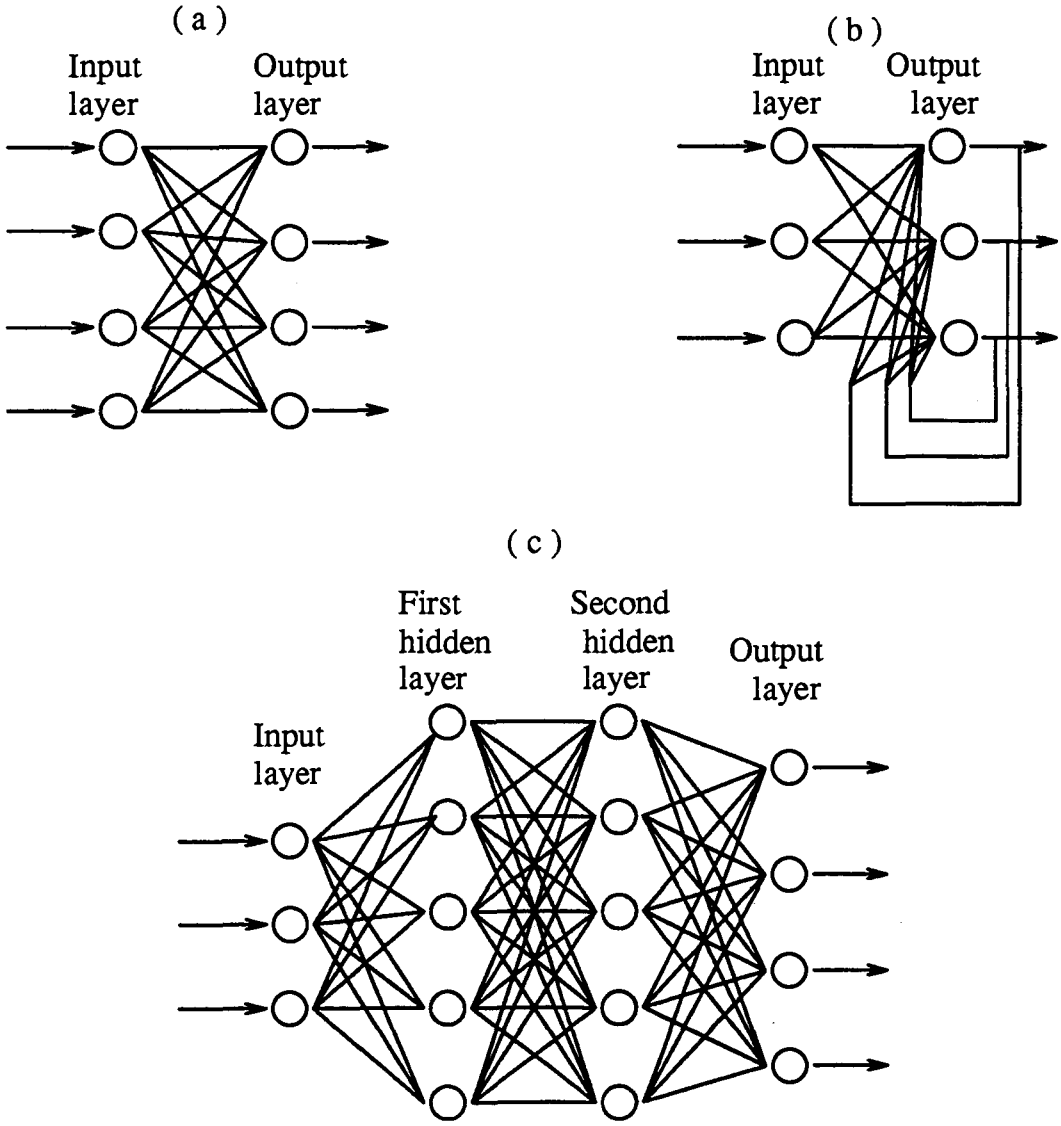


Figure 3.4: Several neural networks
(a) Single-layer network
(b) Single-layer network with feedback
(c) Three-layer network

3.3 Multilayer Feedforward Neural Networks

In this thesis, multilayer feedforward neural-networks with the backpropagation learning algorithm are selected for the design of FDI system. This is due to their capabilities of generating arbitrarily complex decision regions and separating the meshed classes. Moreover, they also have been successfully used in a number of domains such as speech synthesis and recognition, visual pattern recognition and fault pattern classification. As a result, the architecture and design of the multilayer feedforward neural networks, and the backpropagation algorithm will be emphasized in the remainder of this chapter.

3.3.1 Network Topology

The general architecture of the multilayer feed-forward neural networks is depicted in Figure 3.5. Neurons are distributed in three layers: input, hidden and output. Each node is fully connected to the preceding layers. The neurons are interconnected through unidirectional feed-forward communication links. Each link has a weight associated with it. Weights can be positive or negative real values indicating an excitatory or inhibitory effect on a node. Furthermore, each hidden and output node is also connected to a bias which gives a threshold for the activation of the neuron. The bias provides a threshold for the neuron and is essential in order to classify network input patterns into various subspaces.

In the input layer, each node obtains signals from an external stimulus and its value is fed into the hidden layer. Each node in the hidden layer uses the activation function to compute its output and sends the output to a subsequent layer.

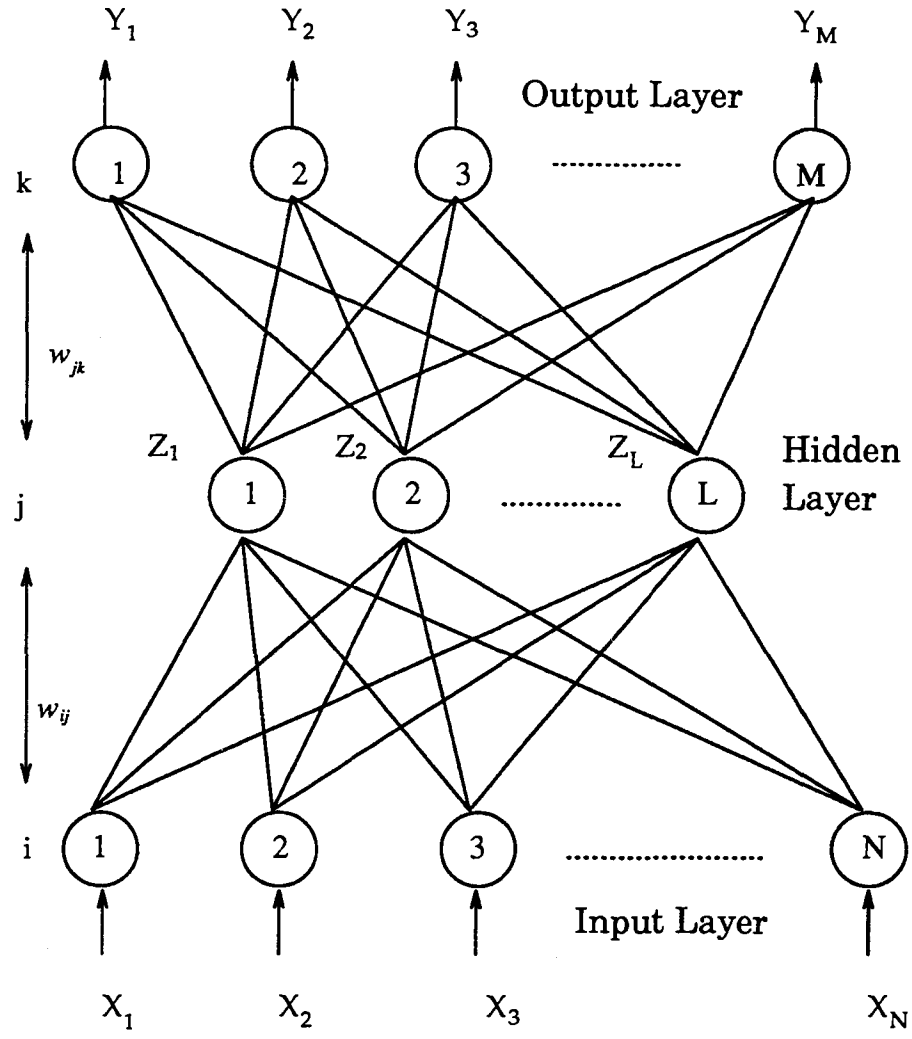


Figure 3.5: Multilayer Feed-forward Neural-nets

The hidden and output nodes carry out two calculations, first a weighted sum of the inputs is taken and then the output is calculated using a sigmoid function. In this case, the outputs of the hidden nodes are calculated from the following formula

$$z_j = f(\sum_{i=0}^{N-1} w_{ij}x_i + \theta_j) \qquad for \quad j = 0, 1, ..., L-1. \tag{3.3}$$

where N and L denote the number of the input and hidden neurons, respectively, x_i is the inputs of the network, z_j the outputs of the hidden layer, w_{ij} the weights between

the input and hidden layer, θ_j the internal biases in the hidden nodes and f is the activation function. The outputs of the network are calculated from the formula

$$y_k = f(\sum_{j=0}^{L-1} w_{jk} z_j + \theta_k) \qquad for \quad k = 0, 1, ..., M-1. \tag{3.4}$$

where M denotes the number of output neurons, w_{jk} is the weights between the hidden and output layer and θ_k is the internal biases in the output layer.

Use of a hidden layer allows the function calculated by the network to not be restricted to simply calculating some linear combination of the network inputs. Since there is no feedback between layers, this architecture has the effect of providing a nonlinear mapping between the input nodes and output nodes. This mapping is completely determined as long as the weights are fixed.

3.4 The Backpropagation Learning Algorithm

Basically, artificial neural networks learn patterns of activation. The network of Figure 3.5 learns by adjusting the weights associated with the connections between the nodes of the network in accordance with a learning rule. In addition, learning is also capable of allowing all the nodes to accomplish the desired state of activation for a given pattern of inputs. The backpropagation algorithm (BP) used in the present work is the most investigated supervised learning algorithm. This algorithm has been implemented in a large number of applications: speech recognition, handwriting classification, prediction and so on. For more detailed information about the BP learning algorithm, please refer to Appendix B.

CHAPTER 4

NEURAL BASED FDI CONTROL SYSTEMS

Fault detection techniques using mathematical models of the process rely on the determination of discrepancies occurring in the process due to the presence of a failure, compared to the normal operating condition of the system. In a nuclear power system, control actuator failures may greatly degrade the performance of the system. Sensor failures may cause abrupt parameter changes in the output matrix, or discrepancies in measurement. In this work, the actual measured values of the output variables are compared to the values of variables obtained from the fault-free plant model. The difference between these variables are called residuals. These residuals are analyzed in the feature extraction to compute new residuals that remove redundant fault patterns. The new residuals are then sent to the input nodes of the failure detection and diagnosis neural network for the classification of various failures. A pressurized water reactor (PWR) of H. B. Robinson's nuclear plant is used as a case study for the

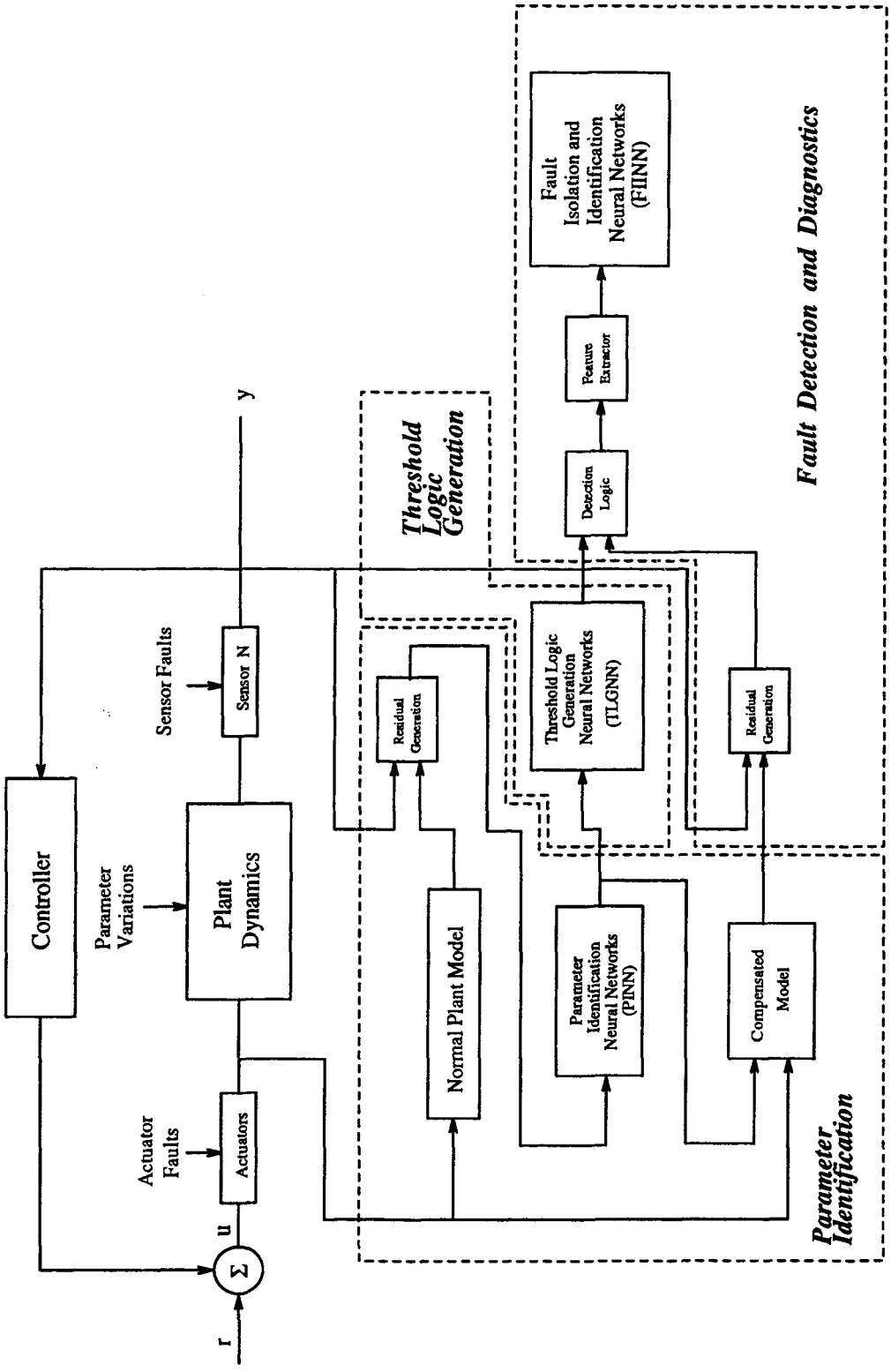


Figure 4.1: Block diagram of the NFDI scheme

development of the neural based FDI (NFDI) system. A computer program, written in Matlab[53], was developed to simulate the dynamics of the PWR-typed nuclear power plant. The program is capable of simulating the plant performance under different parameter variations and fault conditions, and generating necessary data to train and test the designed neural networks. The software package used to simulate the neural networks is "Parallel Distributed Processing" (PDP) by Rumelhart and McClelland[40]. The weights of the network are adjusted using the famous back-propagation algorithm. All data are normalized between 0 and 1 since the neurons of this version of back-propagation used are only able to output value in this range. The network stops training when the mean square error of the whole training set reachs a desired minimun. The plant dynamics and the neural networks simulations are both carried out in Sun SPARCstation IPX.

The design of the NFDI system can be divided into three parts: parameter identification, threshold logic generation, and failure detection and diagnostics . Figure 4.1 illustrates the block diagram of the NFDI scheme.

4.1 The PWR Plant Model

The dynamics of the nuclear power plant is presented in this section without any attempt to get into the detail and physics of the process. A good review of the neutronic process is given in[24]. In [25] the mathematical model of the H.B. Robinson plant was derived and validated by comparison between the theoretical and experimental results. Furthermore, a set of linear coupled differential equations described the H.B. Robinson PWR system are also given in Appendix A.1.

For analysis and design of a control system, the equations describing the dynamics of the PWR are placed into state space formulation. In the following state space representation some simplifying assumptions are made [25,41]. These are (1) the original six groups of delayed neutron decay constants are used to obtain a single delayed neutron group based on the weighted harmonic means of the original six delayed neutron group; (2) by assuming that the pressurizer is large enough to accommodate the steam generator disturbances, its dynamic is neglected in the model formulation; and (3) since the control rods are mainly used for power level adjustment, is is assumed that no power level change occurs so that the term containing $\delta\rho_{rod}$ is dropped. Thus the model of the PWR plant can be simplified to a single input system. The resulting state space formulation is thus of the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.1}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.2}$$

where $x \in \Re^n$, $u \in \Re^m$, and $y \in \Re^r$ are the state, input, and output vectors, respectively. Also **A**, **B**, and **C** are matrices of appropriate dimension. For the H.B. Robinson PWR the state and control are given by

$$x = [\delta P \ \ \delta C \ \ \delta T_f \ \ \delta T_{c_1} \ \ \delta T_{c_2} \ \ \delta T_p \ \ \delta T_m \ \ \delta P_s$$

$$\delta T_{up} \ \ \delta T_{HL} \delta T_{ip} \ \ \delta T_{op} \ \ \delta T_{CL} \ \ \delta T_{Lp}]^T \tag{4.3}$$

and

$$\mathbf{u} = \delta \mathbf{W_{so}} \tag{4.4}$$

where

x_1 Deviation of reactor power (δP)

x_2 Deviation of precursor concentration (δC)

x_3 Deviation of fuel temperature (δT_f)

x_4 Deviation of coolant temperature in 1th node (δT_{c_1})

x_5 Deviation of coolant temperature in 2th node (δT_{c_2})

x_6 Deviation of temperature of primary coolant node (δT_p)

x_7 Deviation of tube metal temperature (δT_m)

x_8 Deviation of steam pressure (δP_s)

x_9 Deviation of reactor upper plenum temperature (δT_{up})

x_{10} Deviation of hot leg temperature (δT_{HL})

x_{11} Deviation of primary coolant temperature inlet plenum (δT_{Ip})

x_{12} Deviation of temperature of primary coolant outlet plenum (δT_{op})

x_{13} Deviation of cold leg temperature (δT_{CL})

x_{14} Deviation of reactor lower plenum temperature (δT_{Lp})

u Deviation of steam flow rate to turbine (δW_{so})

The system distribution matrix **A** and control vectors **B** are given in Appendix A.2.

In this thesis, it is assumed that not all the fourteen state variables are available for measurement, and the output vector is given by

$$\mathbf{y} = [\mathbf{I_8} \mid \mathbf{0}]\mathbf{x} \tag{4.5}$$

where I_8 represents an 8x8 identity matrix. As a result, eight output measurements are assumed to be available in the study.

4.2 Parameter Identification

The linearized mathematical model of PWR is used to accomplish the system design of FDIA. However, in deriving this PWR mathematical model, there are various parameters whose values are approximately determined either through empirical formula or by using curve-fitting techniques. The values of the parameters such as the moderator temperature coefficient of reactivity and Doppler fuel temperature coefficient vary with time during the reactor operation[36]. In the PWR plant, the moderator temperature coefficient of reactivity (α_c) is one of the most important parameters. Therefore, sensitivity to this parameter variation is considered for the FDIA system design.

When parameter variation occurs, the system must detect, identify and take corrective measures to deal with plant parameter perturbation effects. Figure 4.1 shows the proposed structure for parameter identification. The actual plant dynamics to be identified are defined by the following equations,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{A_n}\mathbf{x} + [\mathbf{\Delta A}]\mathbf{x} + \mathbf{B}\mathbf{u} \tag{4.6}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{4.7}$$

where $\mathbf{A_n}$, $\mathbf{B}$, $\mathbf{C}$ are constants and precisely known, and ΔA is an unknown matrix that represents system uncertainties . The important source of parameter variations is assumed to be uncertain perturbation which occur in one or more elements a_{ij} of A. In particular, it can be defined as ΔA=Δa_{ij}. In this thesis, Δa_{14} and Δa_{15} are taken into consideration where both elements are affected by the variation of α_c, the moderator temperature coefficient of reactivity.

The task of parameter identification can be accomplished in three stages: residual generation, plant parameter identification and parameter variation compensation.

4.2.1 Residual Generation

Here the difference between the actual output and the output received from the nominal process model is obtained. These residuals can be formulated as:

$$\mathbf{r(t) = y(t) - y_n(t)} \tag{4.8}$$

where

$$\mathbf{y_n(t) = Cx_n(t)} \qquad \mathbf{\dot{x}_n = A_n x_n + Bu} \tag{4.9}$$

Here subscript n denotes values obtained from the nominal plant. They contains information concerned with the quantitative effects due to parameter variations. Ideally, under no variation condition, the actual output will match the modelled output giving a zero residual. However, if the residual is non-zero, parameter variation will be detected. These residuals are then passed through a neural network parameter identifier for the identification of the perturbated parameter.

4.2.2 Plant Parameter Identification

A neural network identifier is used for parameter identification purposes. It is called *Parameter identification neural networks* (PINN) or PI networks. Figure 4.2 illustrates the specific configuration of the network employed for parameter identification. The network consists of four layers. It has five input nodes corresponding to time, reactor

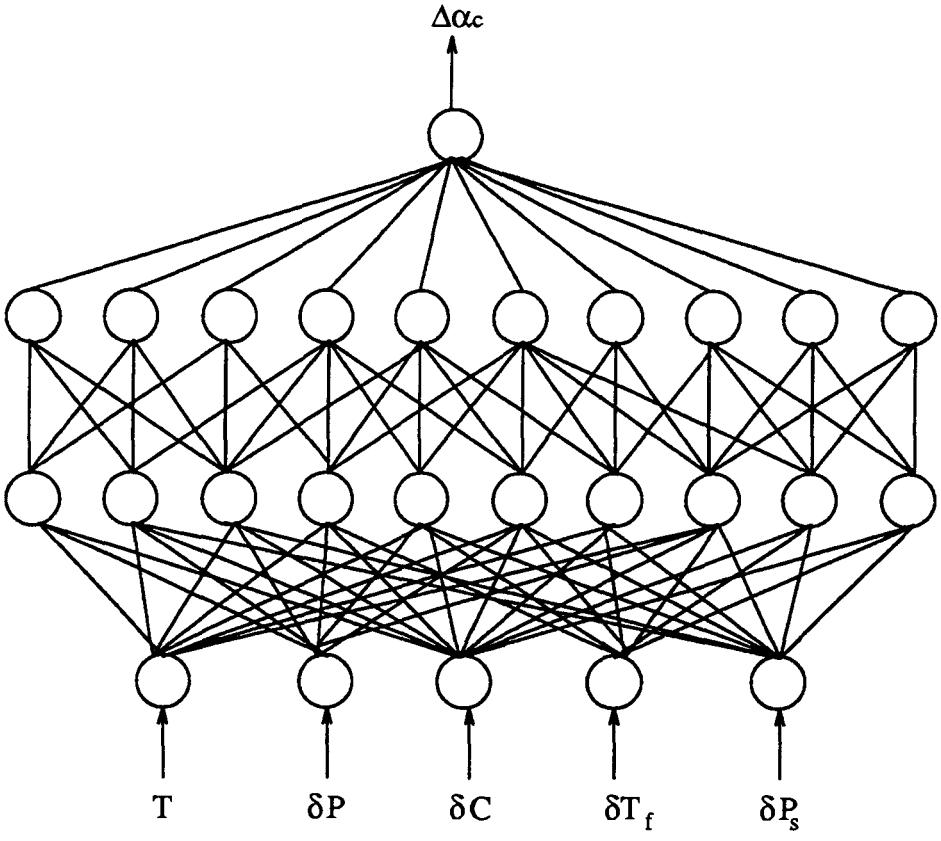


Figure 4.2: PI Networks

power, precursor concentration, fuel temperature and steam pressure residuals. Two hidden layers with twenty nodes, and one output node corresponding to the parameter variation of the moderator temperature coefficient of reactivity are utilized in the neural networks. All four aforementioned output residuals are used to train the neural networks to identify system parameter. The reason for using only these four output measurements is that they show the greatest sensitivity to uncertainty and variation α_c. A range of variations of the parameter are simulated to generate the training data. This parameter is assumed to vary within $\pm 2\%$ of its nominal value.

The training data were generated under nine different fault-free situations for the

process in the transient as well as steady state. The time interval for identification was set to be 30 seconds and the sampling interval for measurement is 0.1 second. As a result, the training data contains 2700 measurement patterns corresponding to the four measurable process variables which were introduced to the network for identification. An example of the training sets is given in Table 4.1, which shows the unnormalized training pattern used at t=10 & 10.1 seconds.

In the table, the values of the four measurable variables and time are used as the input teaching data. Each output pattern corresponds to the magnitude of parameter variation. All measurements are scaled to a continuous range from 0 to 1. The scaling makes the classification easier to learn, because the original measurement data contains both small and large values.

For example, the residuals of deviation of normalized precursor concentration is about 17790.11 and the residuals of the deviation of reactor power is -10.8360. Without the scaling, the fluctuations in large measurement values would dominate the operation of neural networks. The weight of the network is adjusted using the backpropagation algorithms. It stops training when the mean square error is smaller than 0.003. The initial weight values are small random numbers in the interval of [-0.1,0.1] and the training coefficients of the backpropagation algorithm are selected so that the learning rate is $\eta = 0.95$ and the momentum term $\alpha = 0.1$. Other values of the coefficients have also been simulated and these are selected because they yield fast learning.

A number of neural networks of various sizes were also trained for parameter identification. The goal was to find a small network that could be trained to a minimum error. A 80 hidden unit(HU) network was selected to start the training. The training

		Input data			*Output data*
Node 1	Node 2	Node 3	Node 4	Node 5	Node 1
Time	δP	δC	δT_f	δP_s	$\Delta\alpha_c$
10.0	-10.8360	17790.11	2.0598	1.5648	2%
10.0	-6.2808	14686.25	1.7627	1.8763	1.5%
10.0	-2.9273	10692.25	1.3213	1.5084	1%
10.0	-0.8193	5799.58	0.7342	0.7743	0.5%
10.0	0.0000	0.00	0.0000	0.0000	0%
10.0	-0.5119	-6714.46	-0.8826	-0.4750	-0.5%
10.0	-2.3968	-14351.48	-1.9150	-0.2977	-1%
10.0	-5.6962	-22918.46	-3.0983	0.8985	-1.5%
10.0	-10.4506	-32422.51	-4.4337	3.4936	-2%
10.1	-11.4237	17210.21	1.9441	1.0920	2%
10.1	-6.7007	14315.03	1.6856	1.6112	1.5%
10.1	-3.1899	10488.42	1.2765	1.3855	1%
10.1	-0.9402	5720.08	0.7152	0.7364	0.5%
10.1	0.0000	0.00	0.0000	0.0000	0%
10.1	-0.4171	-6681.52	-0.8708	-0.4731	-0.5%
10.1	-2.2385	-14333.88	-1.8988	-0.3173	-1%
10.1	-5.5109	-22966.17	-3.0855	0.8480	-1.5%
10.1	-10.2799	-32587.16	-4.4324	3.4184	-2%

Table 4.1: Measurement patterns used for training PI Networks

was then reduced to 6 hidden unit network. The smallest network that achieved a E_p of 0.03 was one with 20 hidden nodes. As expected, when the number of hidden units was lowered it became harder for the network to learn.

Table 4.2 gives a summary of the number of hidden units, the lowest E_p achieved, and the number of learning iteration for each network. All of the training runs used the same learning rate and the same momentum aforementioned.

When a hidden layer is used, recognition rate depend on the number of hidden

# of HU	Lowest E	# of learning iteration
80	0.078	33659
40	0.014	26590
20	0.003	29050
10	0.150	29072
6	0.597	29800

Table 4.2: Results of training runs for PI Networks

units present. As a rule of thumb, the greater the number of units, the better the recognition. This is due to a better distribution of "knowledge." Using too many hidden units, however, may increase the complexity of the error surface such that training patterns can not be seperated. Thus, it was concluded that the number of the hidden units must be large enough to form a decision region that is as complex as is required by the given problem, and small enough so that the generalization ability remain good.

4.2.3 Parameter Variation Compensation

After the parameter is identified, its effect on ΔA has to be assessed. Furthermore, since in this case the actual system and the nominal modal could differ by $[\Delta A]$x, this effect need to be compensated for in the model. This can be accomplished by adding a compensatory term $\Delta\hat{\mathrm{A}}$ to the nominal plant model A_n so that the actual system matchs with the fault-free model. Hence, the compensated plant model has the following state variable equation:

$$\dot{\hat{\mathbf{X}}} = \mathbf{A_n}\hat{\mathbf{X}} + \mathbf{BU} + [\mathbf{\Delta\hat{A}}]\hat{\mathbf{X}} \tag{4.10}$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{X}} \tag{4.11}$$

where $\hat{}$ denote values obtained from the compensated fault-free plant model. After parameter accommadation, the actual system outputs and the new model outputs are used to generate residuals to be fed into the failure classifier.

4.3 Threshold Logic Generation

The simple way to detect a failure is to compare the residual magnitudes to a threshold value. However, in an uncertain system the residuals are affected by both failures and parameter variations. The thresholds used must be insensitive to the system uncertainties and sensitive to possible faults. Selecting low threshold increases the numbers of false alarms; using large threshold decreases the effect of fault detection.

After parameter compensation, the error of the plant's measurable outputs between the actual system and the compensated model will be approximately zero under fault-free condition. These differences can be used as threshold values to minimize false alarms and missed detections for various parameter variation situations.

In order to have better detection logic for monitoring both the time of occurence and location of the fault, a neural network can be used to generate threshold values under various parameter variations. Figure 4.3 shows the architecture of the neural network for threshold logic generation. It is called *Threshold logic generation neural networks* (TLGNN) or TLG networks. The network is constructed by four layers. The input layers contains 2, the hidden layer 20, and the output layer 8 nodes. Time and the variation of parameter α_c obtained from the output of the PI networks are fed into the input nodes. The output nodes generate eight different threshold values corresponding to each of the output residuals. Similar to parameter identification, the

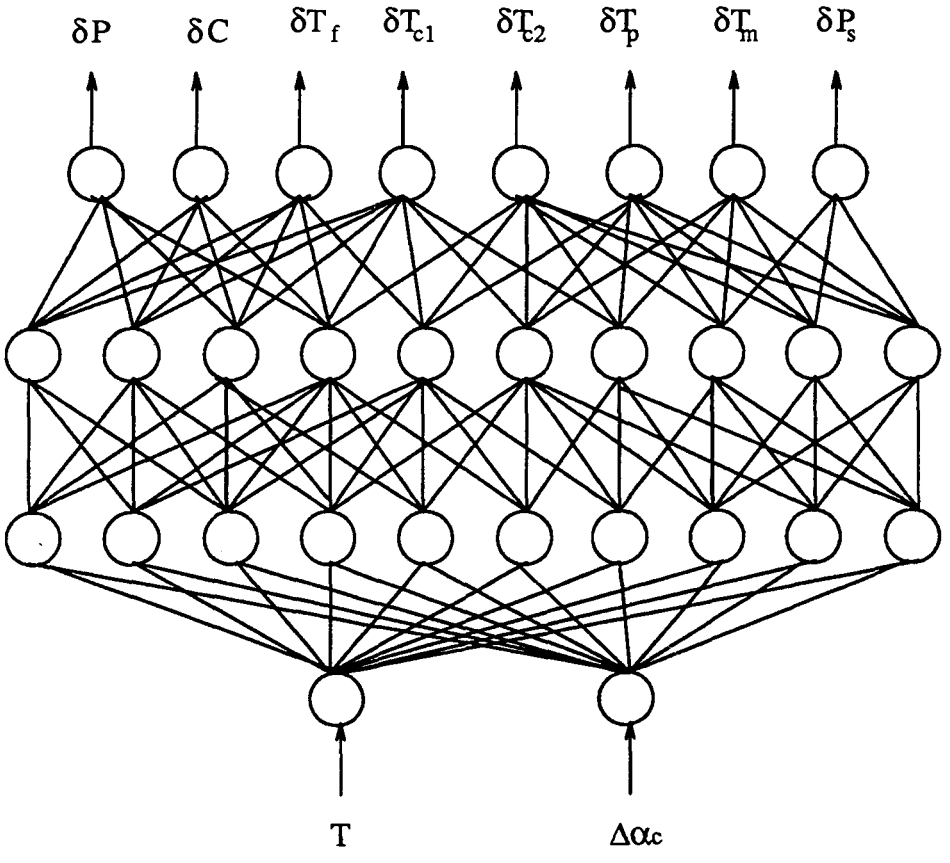


Figure 4.3: TLG Networks

nine different aforementioned variations are used to generate the teaching patterns for the neural networks.

The training data is taken at a steady state as well as transient situation. The simulation was also carried out for 30 seconds using a sampling interval of 0.1 second. 2700 measurement patterns were generate to train the neural networks. Table 4.3 indicates the unnormalized measurement patterns used to train the network at t=10 & 10.1 seconds. These data were normalized between 0 and 1 before training. Each input pattern corresponds to time and the magnitude of parameter variations.

Input data		*Output data*							
n1	n2	n1	n2	n3	n4	n5	n6	n7	n8
Time	$\Delta\alpha_c$	δP	δC	δT_f	δT_{c1}	δT_{c2}	δT_p	δT_m	δP_s
10.0	2%	-0.2070	141.4653	0.0105	0.0028	0.0025	0.0039	0.0010	-0.0381
10.0	1.5%	-0.0196	4.8867	-9.7485	3.5894	3.5593	-3.6716	-9.5860	-0.0114
10.0	1%	-0.0577	123.8976	0.0137	2.7057	3.2672	0.0015	0.0016	0.0086
10.0	0.5%	0.0158	80.8234	0.0115	-8.4967	-6.6039	-3.7231	-0.0004	0.0032
10.0	0%	-0.0036	52.6350	0.0068	-2.8442	-1.9578	0.0005	2.2036	0.0027
10.0	-0.5%	0.0158	80.8234	0.0115	-8.4967	-6.6039	-3.7231	-0.0004	0.0032
10.0	-1%	-0.0250	-127.7671	-0.0164	0.0013	9.1037	0.0013	0.0012	0.0070
10.0	-1.5%	0.0160	14.0035	0.0032	-4.4248	-3.6644	-6.7337	-7.4303	-0.0036
10.0	-2%	-0.1698	-444.5725	-0.0596	0.0060	0.0044	0.0086	0.0105	0.0886
11.0	2%	-0.2096	48.9444	-0.0055	0.0036	0.0033	7.8708	-0.0045	-0.0938
11.0	1.5%	-0.0082	-0.8750	-0.0018	2.1590	2.3616	-4.6938	-6.7626	-0.0064
11.0	1%	-0.0700	89.3959	0.0071	7.9124	8.0839	8.5541	4.2133	-0.0044
11.0	0.5%	0.0056	78.7433	0.0103	-6.2881	-4.4241	-2.8979	0.0005	0.0039
11.0	0%	-0.0086	46.1898	0.0052	-1.2547	-0.0004	0.0001	1.5712	0.0023
11.0	-0.5%	0.0056	78.7433	0.0103	-6.2881	-4.4241	-2.8979	0.0005	0.0039
11.0	-1%	-0.0220	-127.2620	-0.0153	0.0012	7.9466	0.0012	9.2497	0.0037
11.0	-1.5%	0.0153	18.8003	0.0037	-4.6323	-3.6138	-5.9949	-6.6448	-0.0042
11.0	-2%	-0.1773	-478.2476	-0.0617	0.0062	0.0045	0.0084	0.0100	0.0813

Table 4.3: Unnormalized measurement data used for training TLG Networks

# of HU	Lowest E	# of learning iteration
100	0.0740	54567
60	0.0654	67987
20	0.0589	55456
10	0.1042	56685
4	0.9371	54500

Table 4.4: Results of training runs for TLG Networks

A various number of hidden units were also explored to find the network architecture that gives the least error during training. Different numbers of learning iterations were also investigated. A learning rate η and a momentum term coefficient α of 0.9 and 0.6, respectively, were chosen for this study. Table 4.4 shows how the performance of the network under a various numbers of hidden units.

With the aid of neural networks, the thresholds are set such that the detection scheme is able to distinguish between parameter variation and faults, therefore enhancing the accuracy of the fault detection process.

4.4 Failure Detection and Diagnostics

In the most of the FDI techniques, the estimator's innovation process is monitored for jumps which could indicate the presence of a fault. Once a fault is detected, it proceeds to a decision making process for fault identification and isolation. In this thesis, the failure detection and diagnostic system consists of three major parts: residual generation, feature extraction and failure isolation and identification neural networks. The block diagram of this system is illustrated in Figure 4.1.

4.4.1 Residual Generation

The outputs obtained from the compensated fault-free model are substracted from the actual outputs to form a residual as:

$$\Delta \mathbf{y}(\mathbf{t}) = \mathbf{y}(\mathbf{t}) - \hat{\mathbf{y}}(\mathbf{t}) \tag{4.12}$$

If the compensated and actual measurements generally agree, the residual $\Delta y(t)$ will be approximately zero. If a fault is present, this term will diverge and yield a large residual magnitude. The magnitudes of residual is then compared with threshold value obtained from the TLG networks for failure detection.

4.4.2 Feature Extraction

The feature extraction in the detection and diagnostic system is used to filter out redundant failure information and generate a simple decision space for fault diagnosis. There are eight sensors and one actuator in the PWR system. When single or multiple faults occur in any of the seven sensors (i.e. δP, δC, δT_f, δT_{c_1}, δT_{c_2}, δT_p and δT_m), the fault magnitude of these seven sensors can be easily determined by the magnitudes of the residuals of each output variable. By close inspection of eqn. (A.9) in Appendix A, it becomes clear that the actuator failure would immediately affect the steam pressure P_s. Also, simulation study revealed that the actuator failure would not have an immediate impact on the remaining state variables of the system. However, due to feedback all of the sensor faults would affect the steam pressure. Thus, the effect of the residual of the sensor measuring δP_s contains not only its own fault magnitude, but also the failure signatures of the actuator and the other remaining sensors. These extra failure signatures of δP_s obtained from the other sensors can be filtered out by substracting an redundant term from each sensor. The redundant term is defined as the product of the influent constant (Γ_{sensor}) and the fault magnitude ($\Delta sensor$) of the sensor. The influent constants of each sensor are obtained from a simulation program that models the behavior of the plant under single fault condition. In the simulation, seven single sensor failures are simulated seperately with a fault magnitude of 1, in

order to investigate their impact on sensor δP_s.

Influent Constant	Sensors						
	δP	δC	δT_f	δT_{c1}	δT_{c2}	δT_p	δT_m
$\Gamma_{sensor}(t)$	0.0072	135.83	3.2e-2	2.4e-2	0.2489	5.5e-3	-0.0137
$\Gamma_{sensor}(t+1)$	0.0218	413.2	9.7e-2	0.7337	0.7572	1.7e-2	-0.0387
$\Gamma_{sensor}(t+2)$	0.0364	689.67	0.1620	1.2246	1.2639	2.8e-2	-0.0565

Table 4.5: The influent constants used for feature extraction

Table 4.5 tabulates the influent constants obtained from the simulation corresponding to each sensor. The formulation of the feature extractor is defined as follows:

$$\begin{aligned}\Delta\tilde{P}_s(t_i) &= \Delta P_s(t_i) - \Delta P(t_i) * \Gamma_{\delta P}(t_i) - \Delta C(t_i) * \Gamma_{\delta C}(t_i) - \Delta T_f(t_i) * \Gamma_{\delta T_f}(t_i) \\ &- \Delta T_{c_1}(t_i) * \Gamma_{\delta T_{c_1}}(t_i) - \Delta T_{c_2}(t_i) * \Gamma_{\delta T_{c_2}}(t_i) - \Delta T_p(t_i) * \Gamma_{\delta T_p}(t_i) \\ &- \Delta T_m(t_i) * \Gamma_{\delta T_m}(t_i) \qquad\qquad i = 1, 2, 3. \end{aligned} \tag{4.13}$$

where $\Delta\tilde{P}_s$ denotes the new residuals after feature extractions, $\Delta()$ is the residual of each sensor and $\Gamma_{()}$ denotes the influent constant of each sensor. When the detection logic declares the existence of any faulty sensor, the feature extractor implements eqn 4.13 immediately to eliminate the redundant terms obtained from the seven aforementioned sensors. As a result, the new residual of the δP_s contains only the failure signatures of its own sensor and the actuator. After the presence of a fault, three sampling data of δP_s residual (i.e. $\Delta\tilde{P}_s(t_1)$, $\Delta\tilde{P}_s(t_2)$ and $\Delta\tilde{P}_s(t_3)$) are used for fault diagnosis purpose. This is because the three samplings data of $\Delta\tilde{P}_s$ contain sufficient failure information for the diagnosis of the actuator. After feature extraction, the new residuals are then passed to the next stage, the fault classifier which is implemented in the form of a neural network.

4.4.3 Failure Isolation and Identification

A neural network classifier is implemented for failure detection and diagnosis tasks. It is named *Fault isolation and identification neural networks* (FIINN) or FII networks. Figure 4.4 illustrates the network architecture used for the fault detection and diagnosis. The neural network is comprised of four layers. The input layer has three input nodes correponding to the output residuals of P_s at times t_k, t_{k+1}, and t_{k+2}, obtained with the aid of a delay line, each delays the signal by one sampling period. Any instance of fault of the consecutive residuals of δP_s (i.e. $\Delta\tilde{P}_s(t_k)$, $\Delta\tilde{P}_s(t_{k+1})$ and $\Delta\tilde{P}_s(t_{k+2})$) are fed to the network. In addition, the network also consists of two hidden layers with a total of 10 nodes, and two output nodes associated with the failure condition of both the δP_s sensor and the actuator.

The neural network fault classifier was trained using a set of fault data patterns obtained from a numerical simulation of the dynamics of the PWR plant. The data generated by simulation covered the specific fault range under consideration (i.e. [0,100]) for actuator and [0,2] for the δP_s sensor. Fifty five measurement patterns were used to accomplish the task of fault classification. Table 4.6 indicates part of the unnormalized measurement patterns utilized to train the network. In the table, three sampling data of δP_s are used as the input teaching pattern. Each output pattern contains the fault magnitude of actuator and sensor δP_s. All data are normalized between 0 and 1 before training.

Various architecture of neural networks with different numbers of hidden units were studied to find a network that provide the smallest error during training. In FII network, the learning rate η and momentum term coefficient α are 0.9 and 0.1 respec

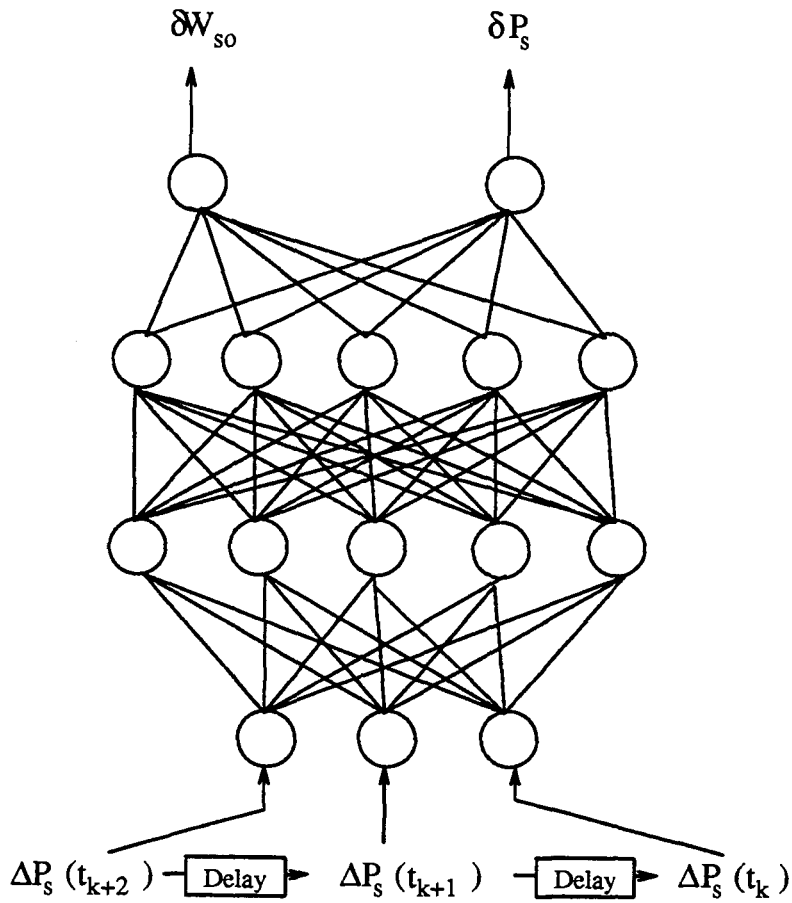


Figure 4.4: FII Networks

-tively. They are selected because they provide quick learning. A various number of learning iterations were also investigated. Table 4.7 lists the performance of the network under a various number of hidden units, different E_p and learning iterations.

The simulation included the effects of all eight sensor faults and a single actuator fault. The network is not only able to classify single and multiple faults, but also capable of estimating the magnitude of the faults simultaneously. The fact that the networks can provide fault magnitude is important since this information can be used for accommodation of the faulty instrument.

Input data			*Output data*	
Node 1 $\delta P_s(t_k)$	Node 2 $\delta P_s(t_{k-1})$	Node 3 $\delta P_s(t_{k-2})$	Node 1 Actuator	Node 2 δP_s
-0.0544	-0.1628	-0.2954	25.0000	0
0.1457	0.0398	-0.0828	25.0000	0.2000
0.3458	0.2425	0.1297	25.0000	0.4000
0.5460	0.4451	0.3423	25.0000	0.6000
0.7461	0.6478	0.5548	25.0000	0.8000
0.9462	0.8504	0.7674	25.0000	1.0000
1.1464	1.0531	0.9799	25.0000	1.2000
1.3465	1.2557	1.1924	25.0000	1.4000
1.5467	1.4583	1.4050	25.0000	1.6000
1.7468	1.6610	1.6175	25.0000	1.8000
1.9469	1.8636	1.8301	25.0000	2.0000
-0.1088	-0.3256	-0.5908	50.0000	0
0.0913	-0.1230	-0.3782	50.0000	0.2000
0.2914	0.0797	-0.1657	50.0000	0.4000
0.4916	0.2823	0.0469	50.0000	0.6000
0.6917	0.4850	0.2594	50.0000	0.8000
0.8918	0.6876	0.4720	50.0000	1.0000
1.0920	0.8902	0.6845	50.0000	1.2000
1.2921	1.0929	0.8971	50.0000	1.4000
1.4922	1.2955	1.1096	50.0000	1.6000
1.6924	1.4982	1.3222	50.0000	1.8000
1.8925	1.7008	1.5347	50.0000	2.0000

Table 4.6: Measurement patterns used for training FII Networks

# of HU	Lowest E	# of learning iteration
40	0.0140	55734
20	0.0094	60342
10	0.0036	52394
8	0.0325	59143
4	0.0904	55249

Table 4.7: Results of training runs for FII Networks

CHAPTER 5

The NFDI-PWR SYSTEM PERFORMANCE

This chapter presents simulated accounts of PWR failure detection and diagnosis performed by the neural based FDI (NFDI) system. The reactor is that of the H. B. Robinson nuclear power plant located in North Carolina which produces 2200 MW(th) at full power. The dynamics of the nuclear power plant derived by Kerlin[24,25] is described in appendix A. Furthermore, the moderator temperature coefficient of reactivity is taken into account for the PWR plant parameter variation since it varies greatly with time during the reactor operation.

The static projective suboptimal controller, designed by Saif[41,42], is used for the control of the PWR system. The projective controller, a class of output feedback controllers, is sought to place the poles at the desired locations, and at the same time approximately minimize the cost functional subject to (5.11)

$$\mathbf{J} = \varepsilon\{\frac{\mathbf{1}}{\mathbf{2}}\int_{\mathbf{0}}^{\infty}(\|\mathbf{X}\|_{\mathbf{Q}}^{\mathbf{2}} + \|\mathbf{U}\|_{\mathbf{R}}^{\mathbf{2}})\mathbf{dt}\} \tag{5.1}$$

where ε represents the expectation operator, Q and R are weighting matrices of appropriate dimension.

The first step in designing a projective controller involves the design of an optimal state feedback LQR that would place the eigenspectrum of the system at desired locations. The resulting closed-loop eigenstructure under a state feedback law will then be used as a reference structure. This reference structure is used to arrive at an oblique projection, and consequently at a projective controller that retains an invariant subspace of the reference eigenspace.

The output control law of the PWR system is given by

$$\mathbf{U^o} = -\mathbf{K^o y} = -\mathbf{K^o C x} \tag{5.2}$$

with the projective controller gain, K°, shown below.

$$\mathbf{K^o} = [3.1697E+2 \;\; -6.0128E-2 \;\; 1.14112E+3 \;\; 1.0677E+4$$

$$1.1019E+4 \;\; 2.4209E+2 \;\; -6.1447E+2 \;\; 2.2941E+1]$$

The PWR system under closed loop control was simulated on a digital computer using a sampling interval of T=0.1 sec. The simulation was carried out for t=30 seconds. The selected test case is an initial impulse disturbance of $\delta T_{Lp}(0)=2^{\circ}F$. It

Failure number	Failure title
1	Single actuator failure under nominal conditions
2	Single sensor failure with a parameter variation of -1.5%
3	Multiple sensor failure with single actuator failure with a parameter variation of 1%
4	All sensor and actuator failure with a parameter variation of 2%

Table 5.1: Failure Scenario titles

should be pointed out that the failure accommodation is successfully accomplished after three sampling periods (the time it takes to detect faulty instruments). To demonstrate the feasibility of the NFDI design, four seperate tests will be conducted in the simulation. These failure scenarios are summarized in Table 5.1.

5.1 Failure No.1: Single Actuator Failure

The first failure scenario involves only actuator failure. In addition, the PWR system is operated under nominal conditions. Therefore, there is no parameter α_c variation (i.e. Δa_{14}=Δa_{15}=0). At t=15 sec, the actuator abruptly increases with a magnitude of 100 (i.e. 10% of the nominal value). This test is intended to indicate that an actuator failure could be detected and identified.

The simulation results are shown in Figure 5.1-4. Figure 5.1 shows that PINN is able to identify the parameter correctly. Since, there is no parameter variation, the values of the elements remain the same (i.e., A_{14}=A_{15}=13700). Figure 5.2 shows the actual thresholds and TLGNN predicted thresholds used for fault detection. Table 5.2 indicates the actual and the desired fault size of the actuator and the sensors estimated by FIINN. The network is able to detect and recognize actuator malfunction immediately with only a 3.531% error. Figure 5.3 illustrates the residuals generated from the actual plant and the compensated fault-free model. At t=15 sec, the actuator fault produces an obvious impact on the residuals of the temperature sensor of primary coolant and tube metal as well as the steam pressure sensor. The effect of the actuator fault on the system outputs is also shown in Figure 5.4.

Case 1 Fault simulated	Identified fault size	Actual fault size
Actuator δW_{so}	96.4693	100
Sensor δP	-0.0026	0
δC	0.0000	0
δT_f	0.0012	0
δT_{c1}	0.0000	0
δT_{c2}	0.0000	0
δT_p	-0.0000	0
δT_m	-0.0020	0
δP_s	0.0370	0
Parameter variation	0%	0%

Table 5.2: Simulation results of FII Networks for Failure No.1

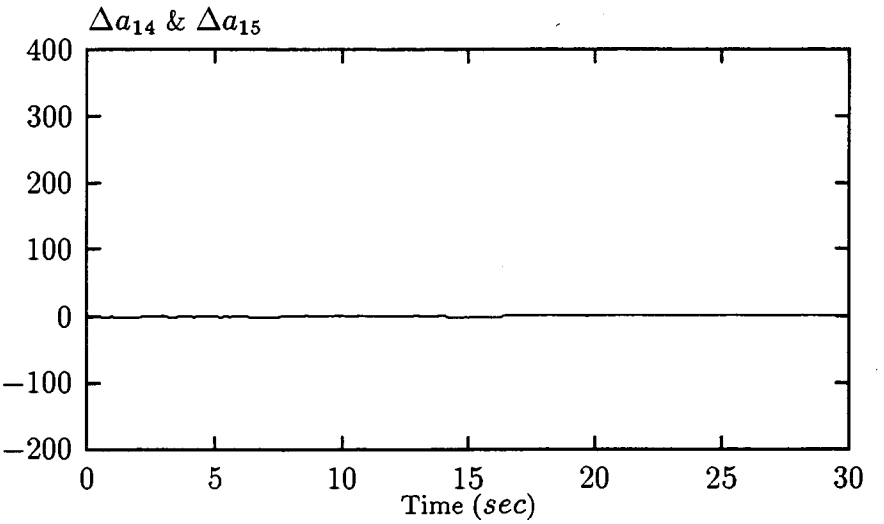


Figure 5.1: Parameter Identification of Failure No.1

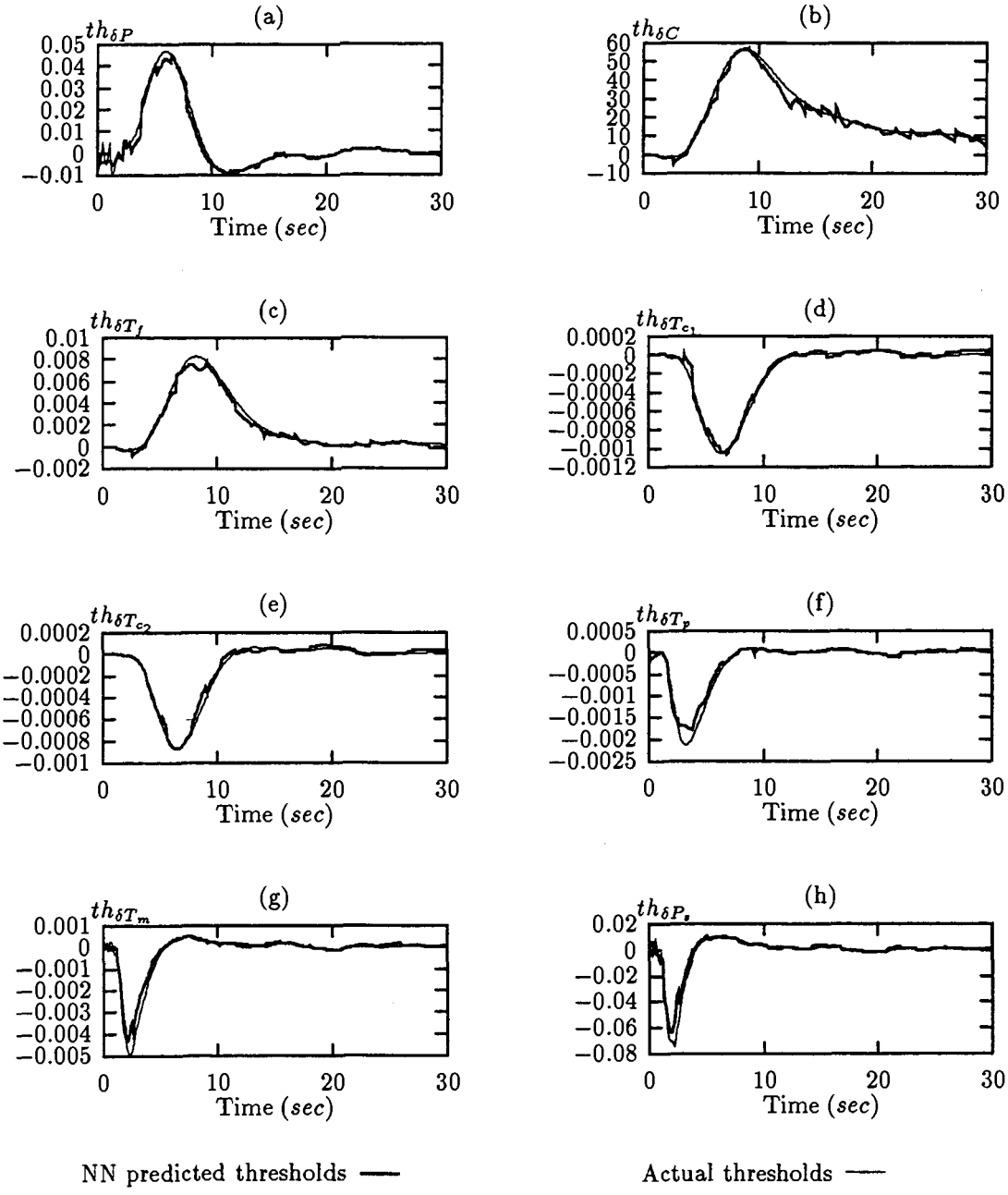


Figure 5.2: The threshold values used for Failure No.1

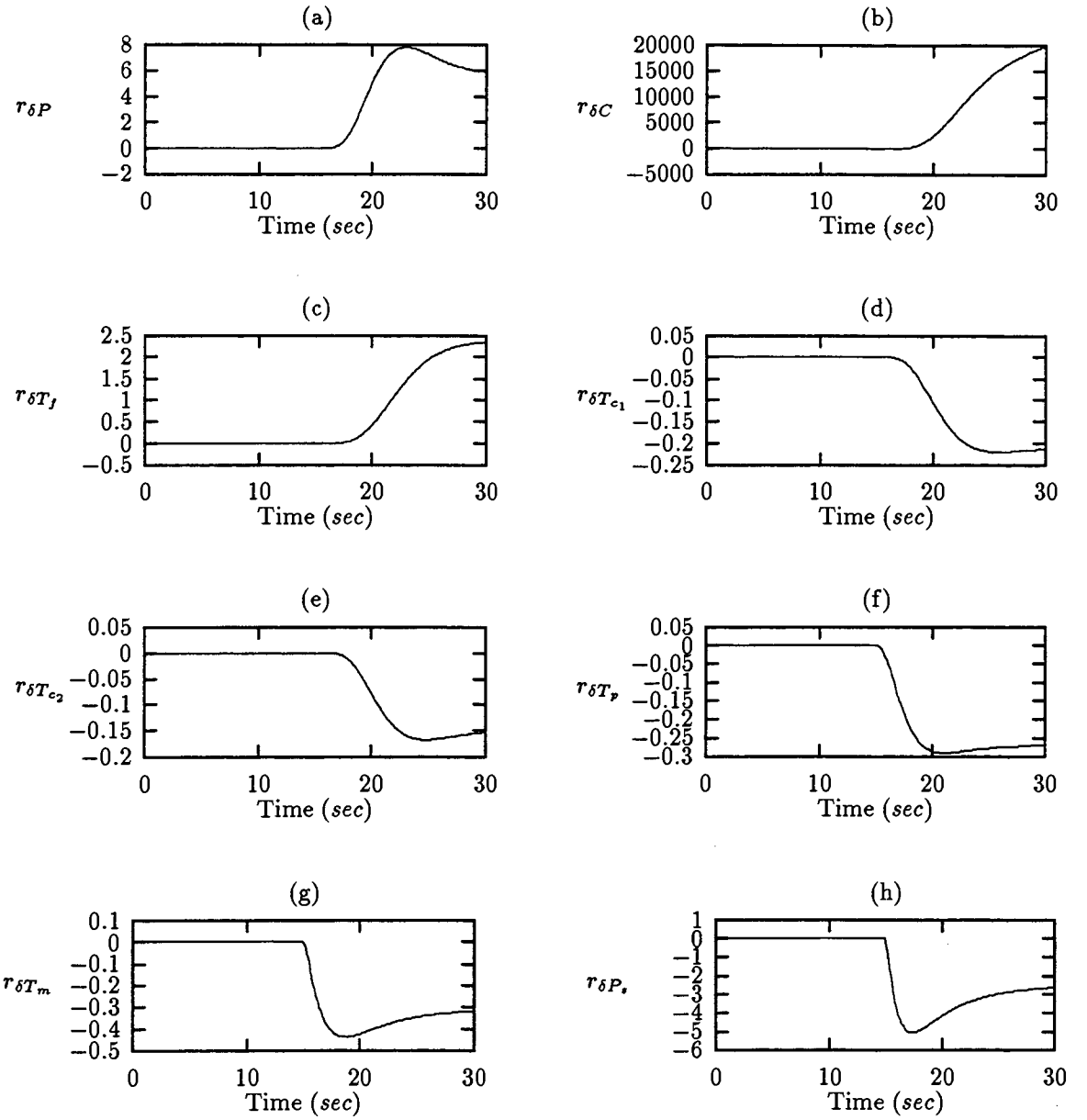


Figure 5.3: The system output residuals of Failure No.1

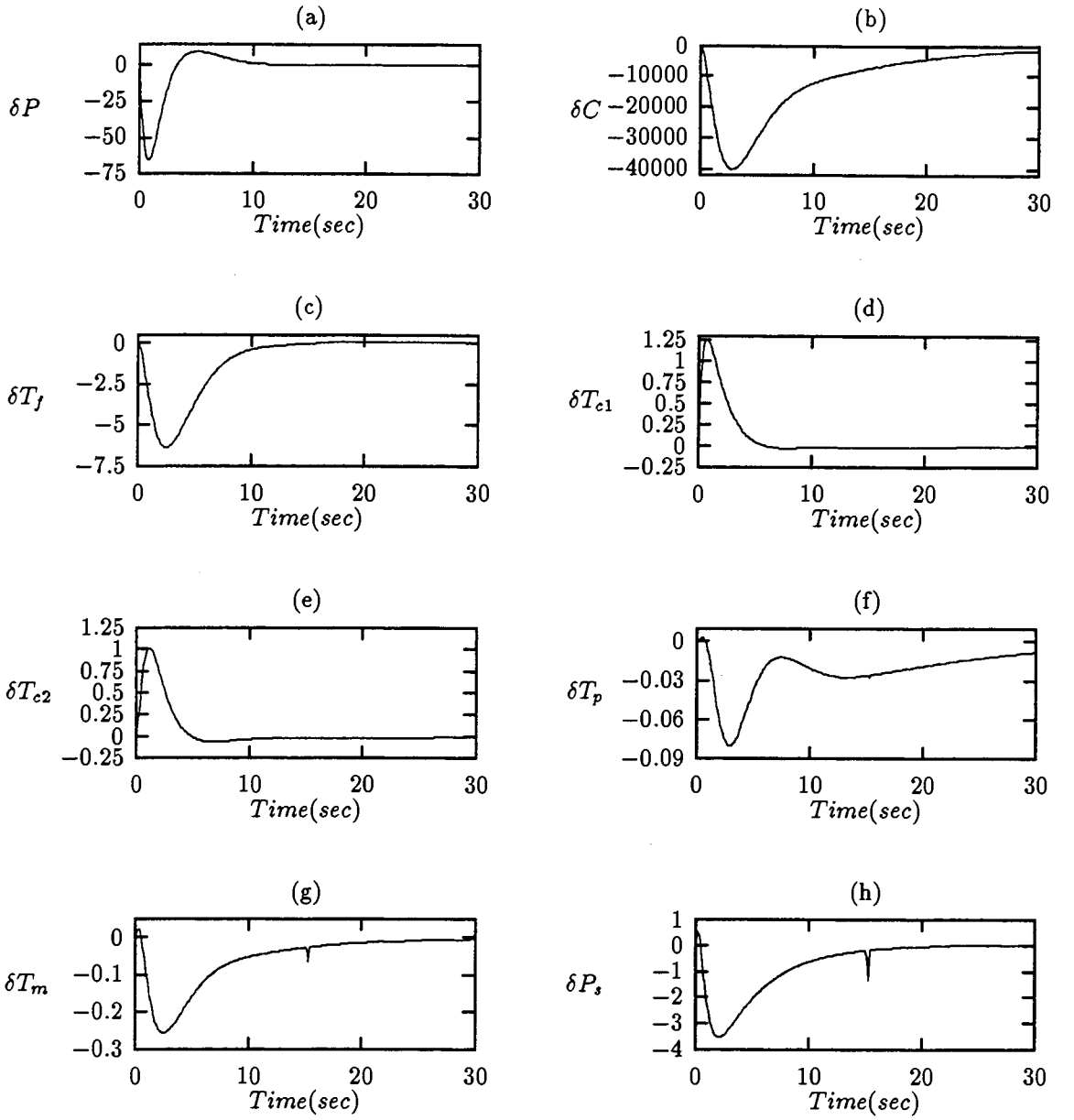


Figure 5.4: The system output responses of Failure No.1

5.2 Failure No.2: Single Sensor Failure

In this failure scenario, it is assumed that a malfunction occurs due to the failure of steam pressure (δP_s) sensor. Furthermore, the moderate temperature coefficient of reactivity is also assumed to have a perturbation of -1.5% of its nominal value (i.e. Δa_{14}=Δa_{15}=-205.5). The sensor fault is assumed to be a step function occurring at t=25 sec with fault size of 0.55. This is to show that the NFDI scheme is capable of detecting and identifying single sensor failures under plant parameter uncertainty.

Figure 5.5 indicates the simulation result of the parameter identification. As shown, it takes about 0.2 sec for PINN to identify the system parameter. The detection logic used for fault detection is shown in Figure 5.6. TLGNN is able to provide much better threshold values under this parameter variation condition. Table 5.3 shows the simulation results of FIINN. It can be observed that FIINN is able to detect and classify the steam pressure sensor failure with an accuracy of 97.22%. The output residuals generated for fault recognition are also shown in Figure 5.7. Figure 5.8 shows the effect of the single sensor fault on the measurable outputs of the PWR plant.

Case 2 Fault simulated	Identified fault size	Actual fault size
Actuator δW_{so}	0.0000	0
Sensor δP	0.0031	0
δC	0.0000	0
δT_f	0.0030	0
δT_{c1}	-0.0002	0
δT_{c2}	-0.0001	0
δT_p	-0.0001	0
δT_m	-0.0001	0
δP_s	0.5653	0.55
Parameter variation	-1.5%	-1.5%

Table 5.3: Simulation results of FII Networks for Failure No.2

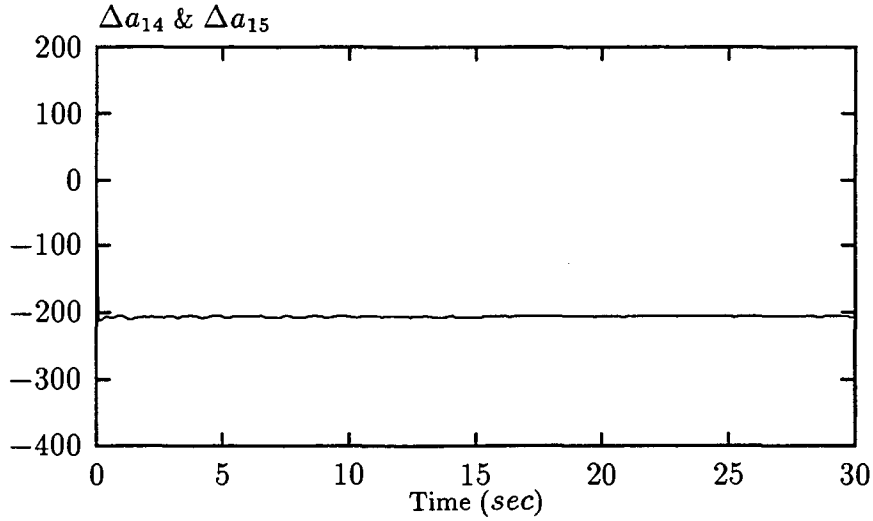


Figure 5.5: Parameter Identification of Failure No.2

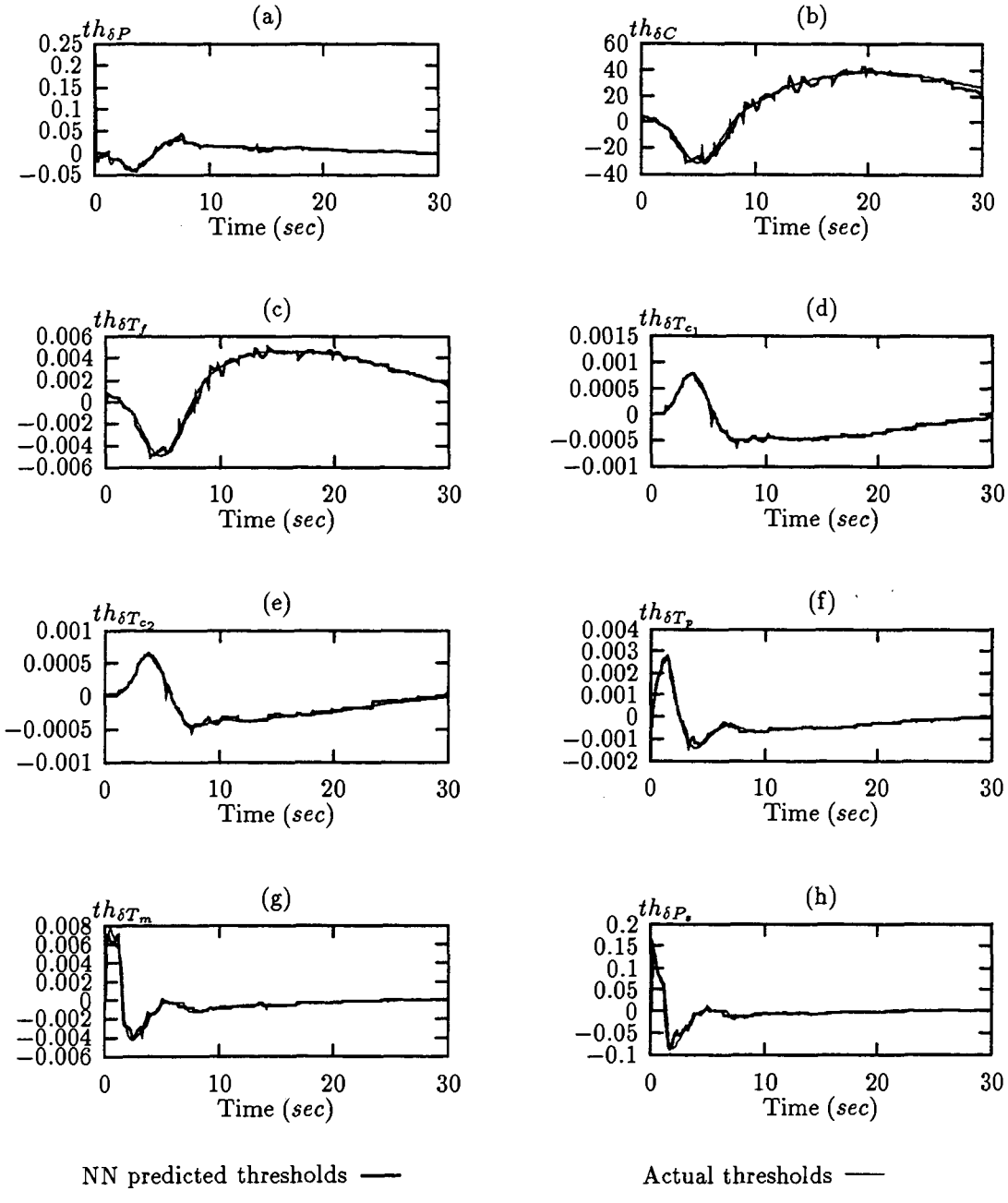


Figure 5.6: The thresholds values used for Failure No.2

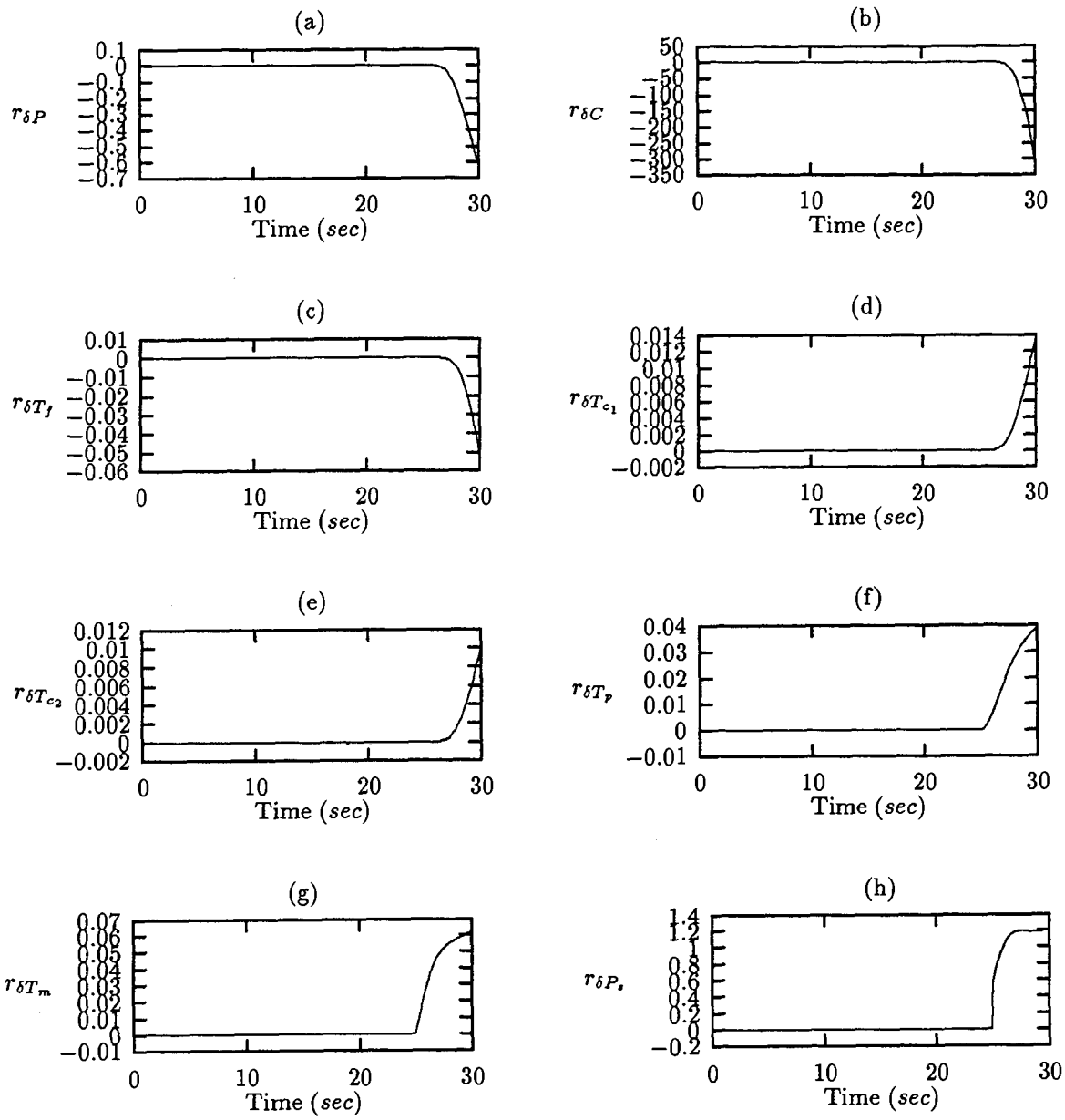


Figure 5.7: The system output residuals of Failure No.2

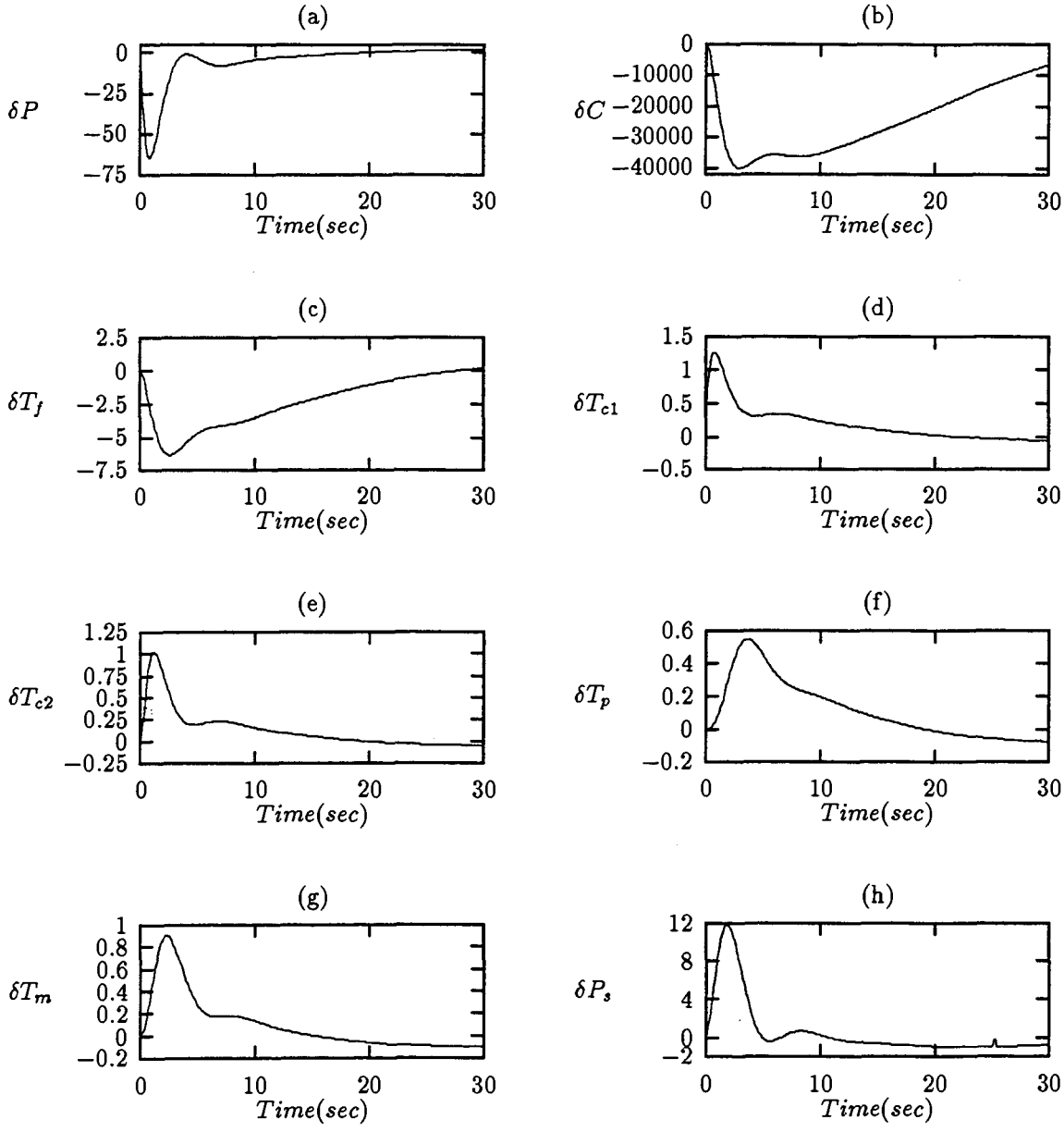


Figure 5.8: The system output responses of Failure No.2

5.3 Failure No.3: Multiple Sensor with Single Actuator Failure

This failure scenario is a combination of four sensor failures and an actuator failure with a parameter α_c variation of 1% from its nominal value (i.e. Δa_{14}=Δa_{15}=137). In particular, the precursor concentration sensor, the temperature sensor of primary coolant node in steam generator, the tube metal temperature sensor and the steam pressure sensor are faulty. The fault size of the faulty sensors are 4500, 0.08, 0.05 and 0.75, respectively. The magnitude of the actuator fault is 50 (i.e. 5% of the nominal value). All the failures are assumed to occur at t=10 sec. The purpose of this test is to illustrate the proposed approach is able to distinguish actuator failure from sensor failures as well as detect and identify them simultaneously.

The system parameter is identified accurately with the support of PINN as shown in Figure 5.9. Figure 5.10 shows the detection logic used for fault isolation and identification obtained from TLGNN. Table 5.4 shows the simulation results obtained from FIINN. At t=10 sec, FIINN can correctly isolate and identify the four sensor and the actuator failures. The output residuals shown in Figure 5.11 and the output trajectories shown in Figure 5.12 can be used to illustrate the behavior of the PWR system in the presence of the faulty instruments.

Case 3 Fault simulated	Identified fault size	Actual fault size
Actuator δW_{so}	49.15	50
Sensor δP	0	0
δC	4620.60	4500
δT_f	0	0
δT_{c1}	0	0
δT_{c2}	0	0
δT_p	0.0814	0.08
δT_m	0.0510	0.05
δP_s	0.7612	0.75
Parameter variation	1%	1%

Table 5.4: Simulation results of FII Networks for Failure No.3

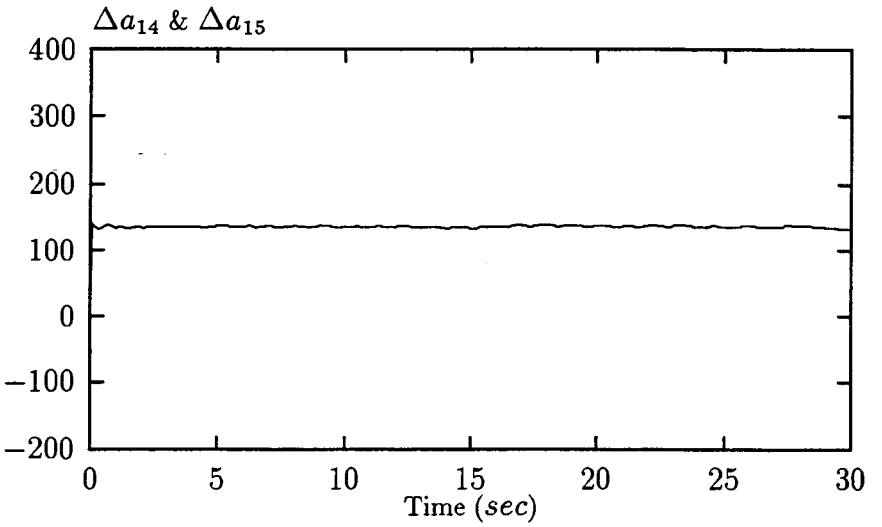


Figure 5.9: Parameter Identification of Failure No.3

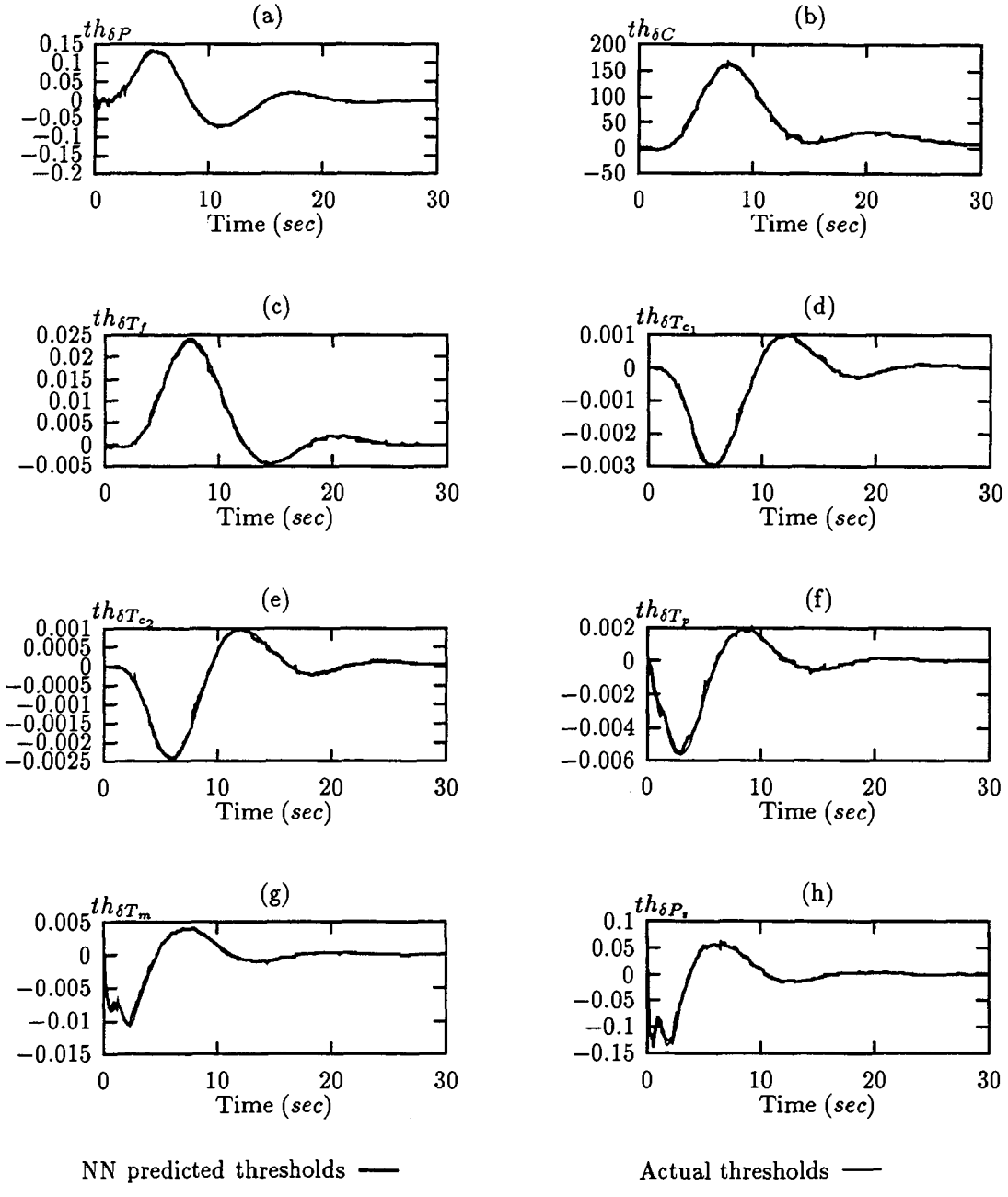


Figure 5.10: The threshold values used for Failure No.3

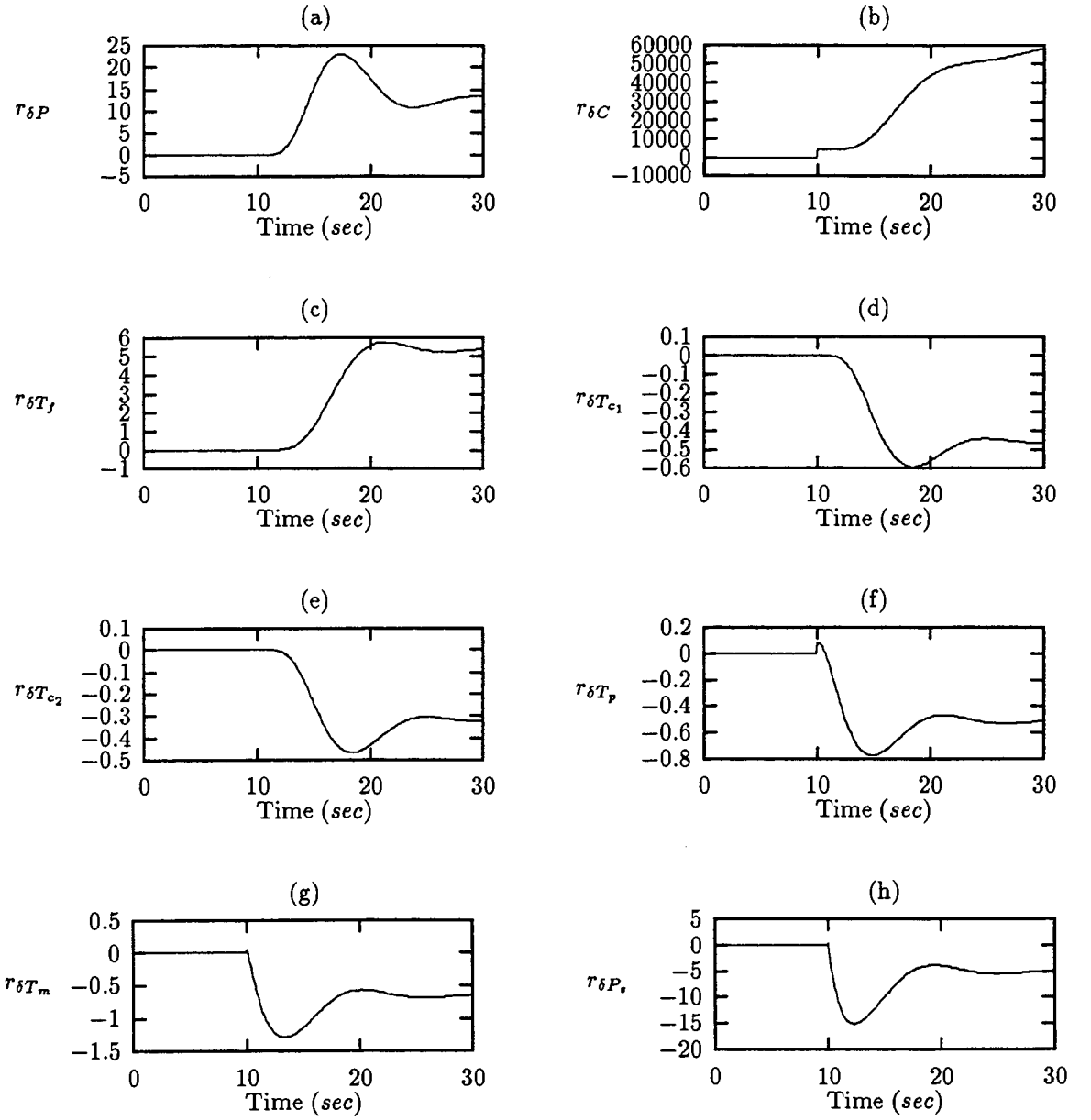


Figure 5.11: The system output residuals of Failure No.3

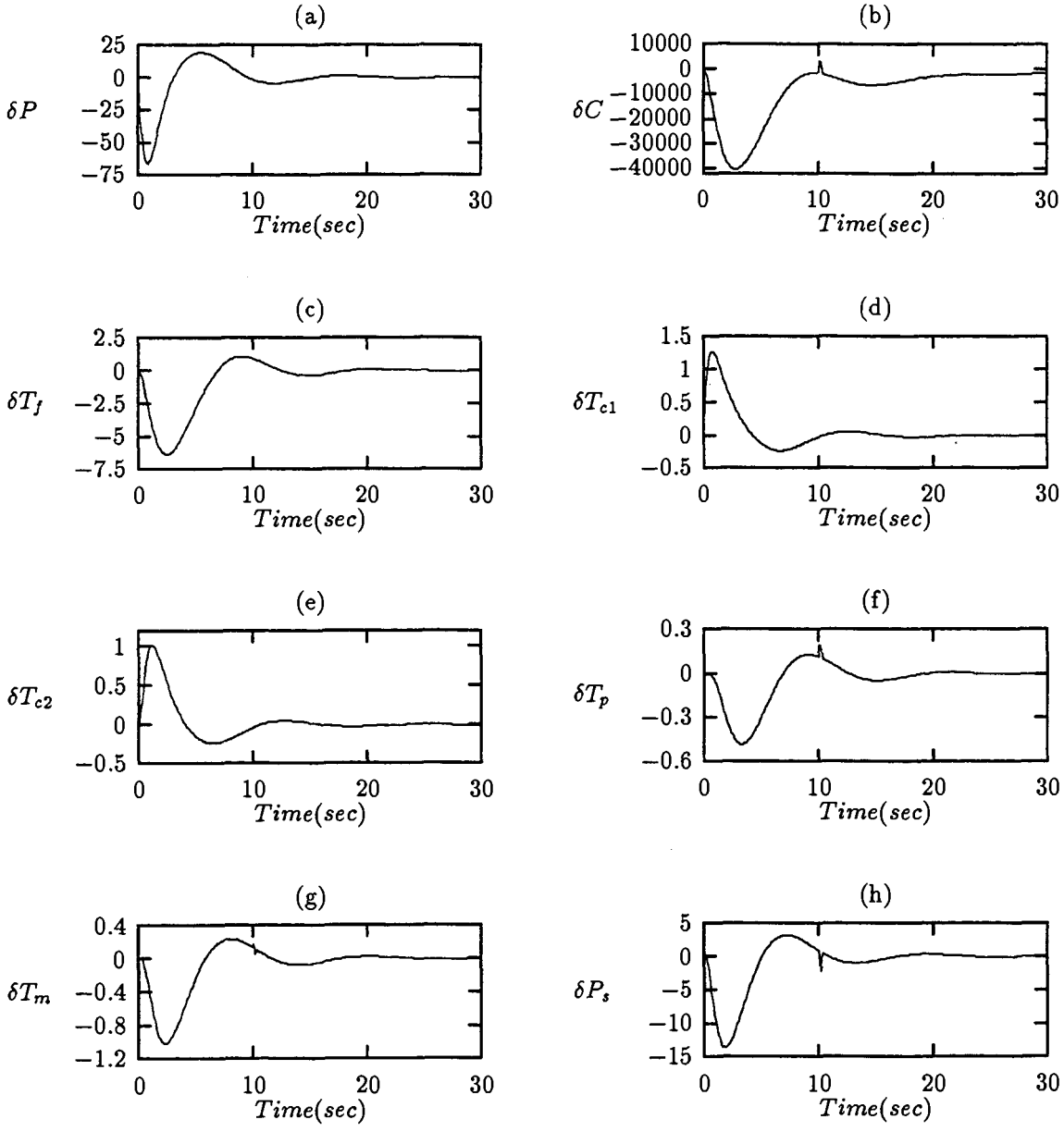


Figure 5.12: The system output responses of Failure No.3

5.4 Failure No.4: All Sensor and Actuator Failure

This failure scenario is similar to Failure No.3, except that in this case, all instruments are faulty. In addition, parameter α_c has a variation of 2% from its nominal value which is considered (i.e. Δa_{14}=Δa_{15}=274). In other words, the malfunctions occur due to the failures of the actuator, the reactor power sensor, the precursor concentration sensor, the fuel temperature sensor, the two coolant temperature sensors, the temperature sensor of primary coolant, the tube metal temperature sensor and the steam pressure sensor. All the actuator and sensors have failed at t=20 seconds. The fault size of the faulty instruments are tabulated in Table 5.5. However, in practice, this may not be the case, it is intended to demonstrate the capability of the detection and diagnostic system.

Figure 5.13 shows the convergence, with respect to time, of the system estimated parameter. It can be seen from this figure that the estimate appear to converge, it takes only 0.2 second for PINN to identify the parameter. After the parameter is correctly identified, TLGNN generate the detection logic for each sensor corresponding to the parameter deviation. Figure 5.14 illustrates the actual and TLGNN predicted threshold values. It is obvious that TLGNN is capable of estimating the threshold values accurately. Table 5.5 shows the result obtained from FIINN. By comparing the actual and desired fault size of the instruments, FIINN demonstrates its capability of classifying the corresponding failures accurately. Figure 5.15 shows the residuals generated for fault detection and diagnosis. At t=20 sec, the residuals of each sensor diverge greatly due to the presence of the failures. Figure 5.16 shows the output responses of the PWR system.

Case 4 Fault simulated	Identified fault size	Actual fault size
Actuator δW_{so}	74.26	75
Sensor δP	3.97	4
δC	3611.44	3500
δT_f	1.5137	1.5
δT_{c1}	1.00	1.00
δT_{c2}	0.4999	0.5
δT_p	0.1030	0.1
δT_m	0.1027	0.1
δP_s	1.5316	1.5
Parameter variation	2%	2%

Table 5.5: Simulation results of FII Networks for Failure No.4

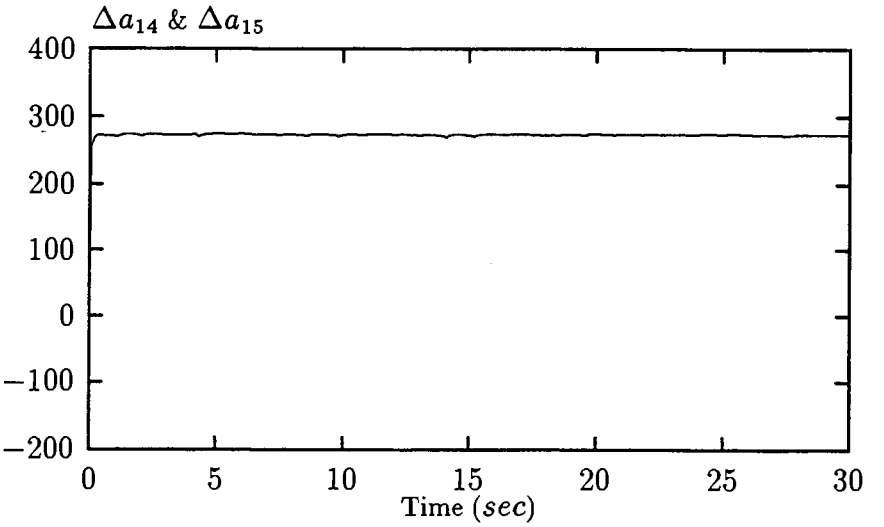


Figure 5.13: Parameter Identification of Failure No.4

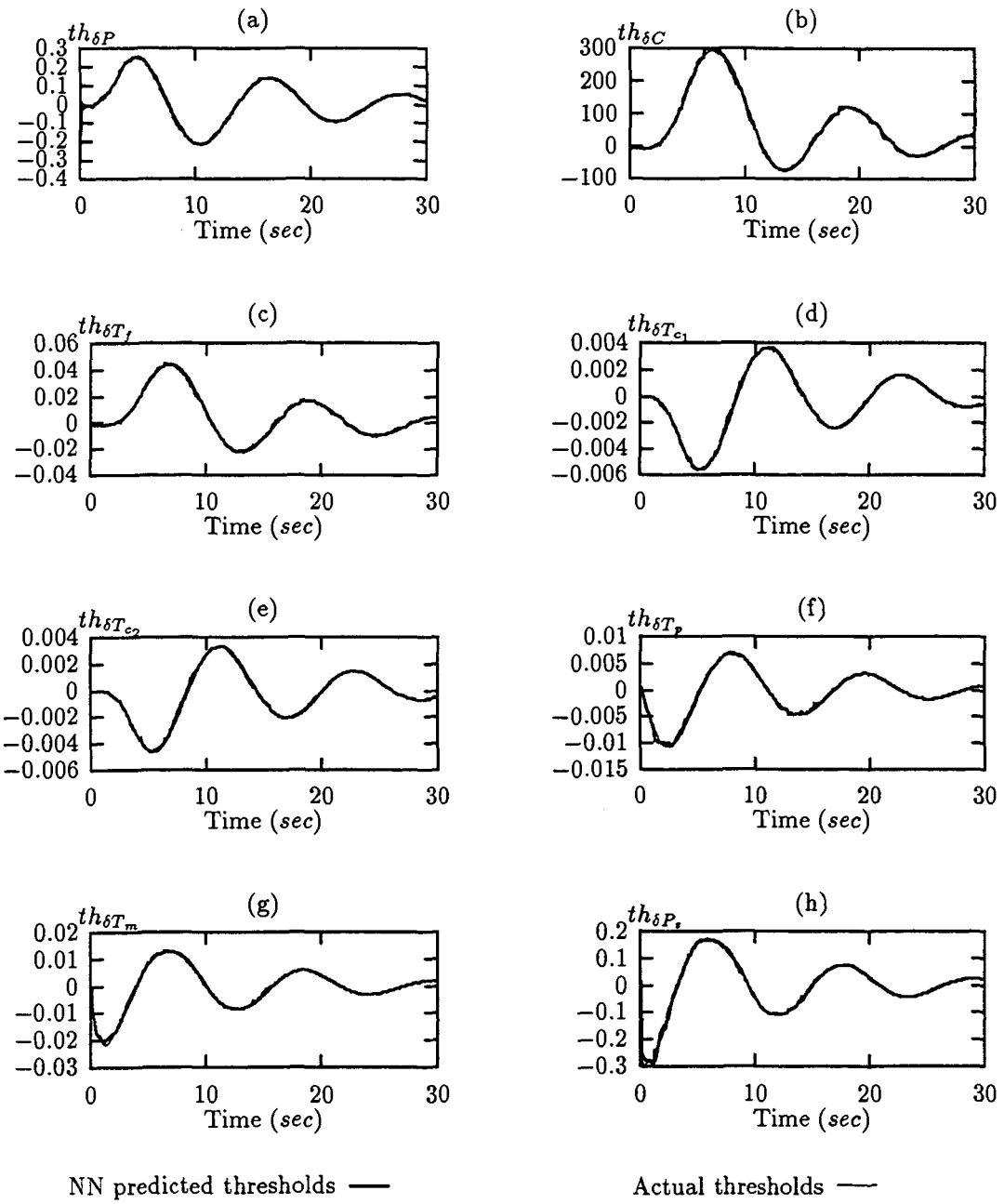


Figure 5.14: The threshold values used for Failure No.4

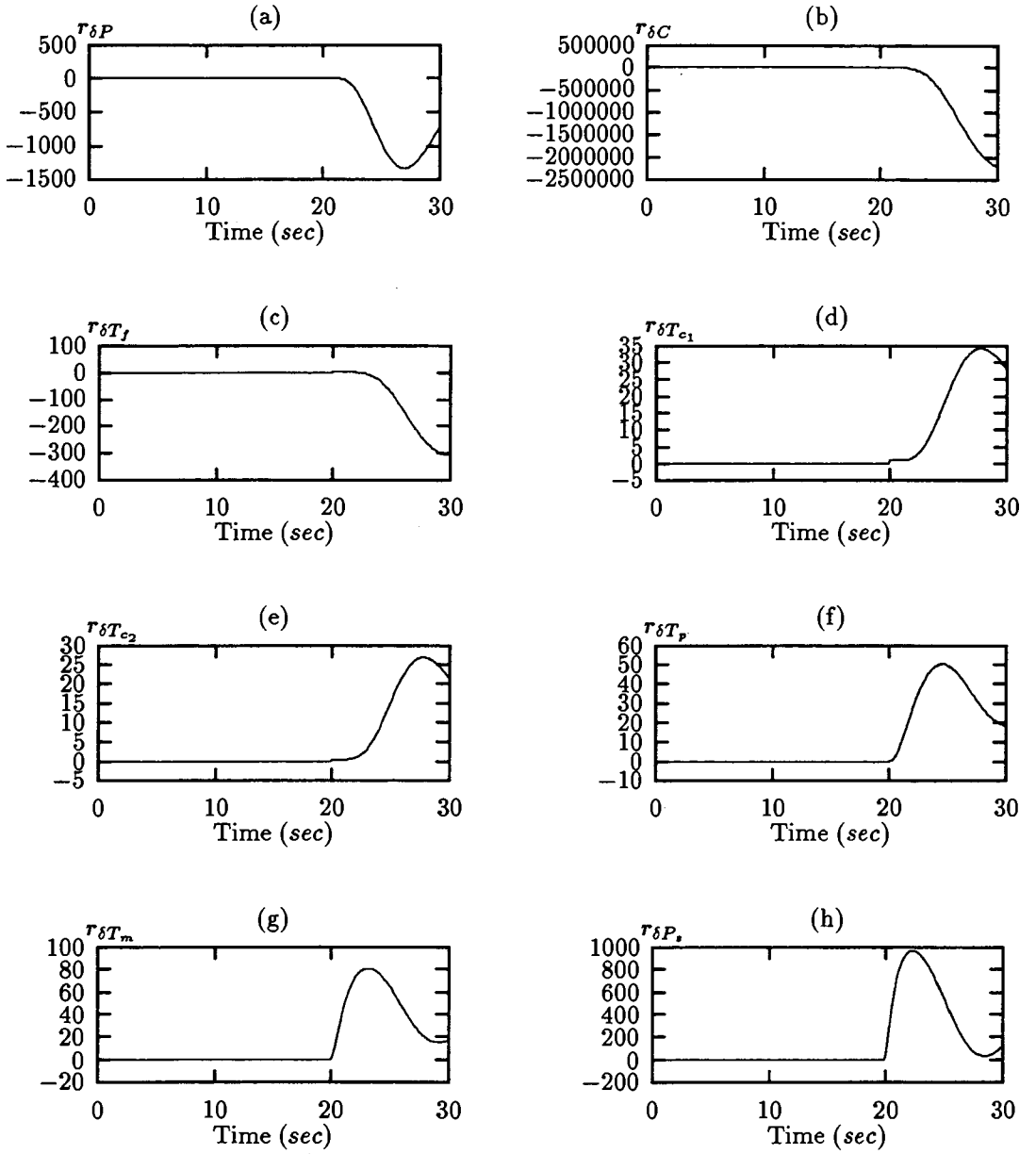


Figure 5.15: The system output residuals of Failure No.4

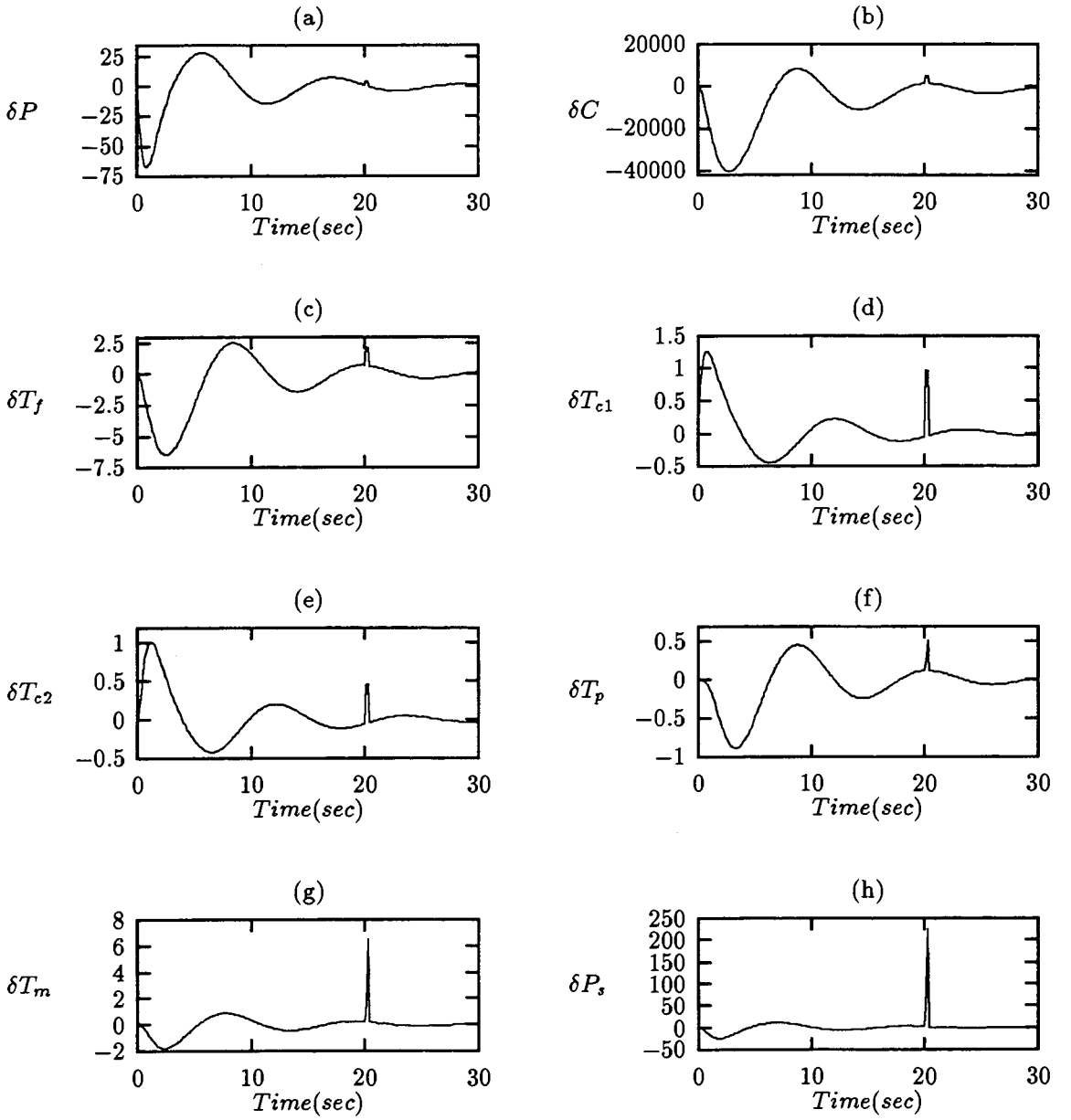


Figure 5.16: The system output responses of Failure No.4

5.5 Summary of Results

An on-line application of neural networks for parameter identification, and fault detection and diagnosis of the PWR-type nuclear power plant was developed and simulated with four different failure scenarios. The NFDI system is capable of identifying the variation of the parameter accurately. Furthermore, the networks are also able to diagnose precisely single and multiple failures. The magnitude of the faults can also be determined correctly. The networks can also detect and diagnose failures occurred at different time. Although failure accommodation was not dealt with here, it can be performed after three sampling periods (the time it takes to detect faulty instruments). Satisfactory results show a promising future for the application of artificial neural networks for fault-tolerant control system design of the PWR-typed nuclear power plant.

CHAPTER 6

SUMMARY AND CONCLUSIONS

This final chapter provides a summary of the thesis and outlines the major conclusions to be drawn.

6.1 Summary

Chapter I motivated the need for fault detection, identification and isolation. A literature review of some traditional and existing methods was briefly described in this chapter. An neural network based design techniques were identified as a promising methodology for FDI.

Chapter II addressed the general ideas of a fault-tolerant control systems. An overview of the neccessary procedures that needed to be taken in accomplishing FDI

were presented, and an dedicated observer scheme and analytical knowledge based method was outlined.

Chapter III introduced artificial neural networks. A brief discussion on the neural network properties was presented. The architecture of multilayer feedforward neural networks was highlighted. The popular backpropagation algorithms was also described.

Chapter IV described the development of a neural network based FDI system. It showed how the neural network and mathematical model based design techniques could be used to achieve FDI. The structure of the networks for parameter identification(PI), threshold logic generation(TLG) and fault detection and diagnosis(FII) were presented.

Chapter V provided simulated accounts of failure detection, identification and isolation performed by the neural based FDI system of the pressurized water reactor(PWR). Details of the simulation were summarized, and the simulation results illustrated that the design approach was able to simultaneously classify single and multiple faults under plant parameter uncertainties.

6.2 Conclusions

In this thesis, an approach based on neural networks and mathematical models for detecting and diagnosing instrument failures was presented. Multilayer neural networks were used at the first level for parameter identification, at the second level for threshold logic generation, and as a pattern recognizer in the third level for fault

detection and diagnosis. Although simulations were used to generate data that were used in training the network to discriminate, it is also possible to collect data about parameter variations, instrument faults and related measurements from the historical databases in the plant and use the historical information to train the network.

This research study makes contributions to the design of a FDI system via neural networks. These accomplishments are itemized below.

- A fundamental problem involving parameter perturbation model-errors for linearlized dynamical sysems has been formulated and a model-error compensator design for the that problem has been described. Neural networks are implemented for identifying the variation of the moderator temperature coefficient of reactivity. The design scheme can quickly detect, identify and compensate for the effects of a broad range of uncertain perturbations in plant parameters. The performance capabilities and structural features of the parameter identification scheme have been illustrated by simulation results.

- Thresholds are usually used to distinguish between a system uncertainty and an instrument fault. Selecting the threshold too low raises the rate of false alarms; selecting it too large decreases the effect of fault detection. A neural network is used to generate threshold values for each sensor under various parameter uncertainties.

- The feature extractor in the fault detection and diagosis scheme is used to filter out redundant failure information and generate a simple decision space. It reduce the training data required for fault classification.

- Neural networks are utilized as a pattern recognizer for actuator and sensor fault

classification. The network is capable of diagnosing correctly in the presence of single and multiple fault operations of actuators and sensors.

Multilayer feedforward networks are selected since they can classify the training data well. The patterns that are not in the training data are also classified correctly indicating the robustness and fault tolerance capabilities of neural networks in diagnostic applications. An increase in the learning rate η or the momentum term coefficient α leads to a higher recall error at convergence. The training cannot be speeded up significantly by changing the learning parameters but using hyperbolic tangent as the nonlinearity in the hidden layer changes the situation dramatically. When a hidden layer is present, recognition rates depend on the number of hidden units present. As a rule of thumb, the greater the number of hidden units, the better the recognition. Using too many hidden units, however, may increase the complexity of error surface such that the training pattern cannot be separated. Therefore, the number of the hidden units must be large enough to form a decision region that is as complex as is required by the given problem. Several kinds of two-hidden-layer networks were examined. Although these network required twice the learning iterations as one-hidden-layer network, the performance was better than that of an one-hidden-layer network since the network is capable of discrminating more complex regions.

In order to draw the effectiveness of the proposed approach, the pressurized water reactor (PWR) of the H. B. Robinson's nuclear plant was used as a test process. Simulation results presented revealed that the design approach was able to detect and diagnose a variety of fault combinations under plant parameter uncertainties. With the aid of the simulations, the proposed NFDI scheme has the following advantage over some of the other existing approaches:

1. The NFDI scheme is capable of detecting the perturbated plant parameter and accurately identifying the magnitude of the variation. In addition, it is also able to compensate the parameter very quickly.

2. The neural based FDI technique can detect and isolate the instrument failures simultaneously as well as identify the size of failures correctly.

3. The proposed approach is able to detect and isolate both sensor and actuator failures. Furthermore, the networks are also able to diagnose precisely in single and multiple system fault scenarios.

4. No observer is required in the proposed NFDI method; therefore, a heavy calculational load for estimation can be avoided. In contrast to other techniques, a dedicated observer scheme(DOS), for example, a number of Luenberger observers were dedicated to each sensors.

5. The NFDI approach has the capability of generating better threshold logic which is insensitive to the plant parameter uncertainties and sensitive to possible faults. As a result, the rate of false alarms can be greatly reduced, which effectively improve the performance of the fault detection.

6. The neural network approach requires no explicit encoding of knowledge as in analytical knowledge-based approaches. Therefore, neural networks appear to be very attractive for applications in which knowledge extraction is difficult or in cases where the interrelationships between process parameters are too complex. Changes in the plant configuration necessitate retraining of the neural network, which is simpler than revising the complex knowledge base of an expert system.

The main conclusions to be drawn is that although a given FDIA problem may be approached in various ways, the proposed neural based FDI approach may prove more useful than conventional methods under certain circumstances. Satisfactory results show a promising future for the on-line application of artificial neural networks for fault-tolerant control systems.

6.3 Towards the Future

The purpose of this study was to understand how neural networks might be effectively used in parameter variation, and fault detection and diagnosis. The preceding accomplishments represent some of the recent progress have been made in this direction. While this work has provided some insight into the neural based FDI system design, it has also raised a number of important questions. The answers to these questions will be left to future researchers. For now some of these questions are briefly itemized as follows:

1. A number of areas where further research is needed is in back-propagation. Guidelines are needed for designing the architecture of a neural network based on characteristics of the process. In addition, more extensive guidelines are also required for setting training parameters to maximize the learning rate.

2. Additional research is required to incorporate more failure modes and parameter variations.

3. Fault accommodation was not considered in this study. It is desired to have system reconfiguration after the presence of a failure in a truly fault tolerant

control system. More work needs to be done in this area.

4. Additional simulation studies are needed, using the failure data generated with the nonlinear simulation to completely verify the validity of the neural based FDI technique.

APPENDIX A

THE PWR DYNAMIC SYSTEMS

A.1 The PWR Mathematical Modeling

The dynamics of the nuclear power plant is presented in the following without any attempt to go into the detail and physics of the process. A good review of the neutronic process is given in[24]. In [25] the mathematical model of the H.B. Robinson's plant was derived and validated by comparison between the theoretical and experimental results. The following set of linear coupled differential equations describe the PWR system:

Core

The point kinetics model with six delayed neutron groups with reactivity feedback

was used to represent the reactor power as given by the following linearized equations:

$$\frac{d\delta P}{dt} = -\frac{\beta}{\Lambda}\delta P + \sum_i \lambda_i \delta C_i + \frac{\alpha_p P_0}{\Lambda}\delta P_p + \frac{P_0}{\Lambda}\delta \rho_{rod}$$

$$+\frac{\alpha_c P_0}{\Lambda}\sum F_{ci}\delta T_{ci} + \frac{\alpha_f P_0}{\Lambda}\sum F_{fi}\delta T_{fi} \tag{A.1}$$

$$\frac{d\delta C_i}{dt} = \frac{\beta_i}{\Lambda}\delta P - \delta\lambda_i \delta C_i, \qquad \forall i = 1, 2, \cdots, 6 \tag{A.2}$$

The following equations use a nodal approximation for fuel and coolant temperatures to develop the core heat transfer model:

$$\frac{d\delta T_{fi}}{dt} = \frac{Q_{fi}}{(MC_p)_{fi}}\delta P - (\frac{UA_f}{MC_p})_{fi}(\delta T_{fi} - \delta T_{c_{1i}}) \tag{A.3}$$

$$\frac{d\delta T_{c_{1i}}}{dt} = (\frac{UA_f}{MC_p})_{ci}(\delta T_{fi} - \delta T_{c_{1i}}) - \frac{2}{\tau}(\delta T_{c_{1i}} - \delta T_{cin}) \tag{A.4}$$

$$\frac{d\delta T_{c_{2i}}}{dt} = (\frac{UA_f}{MC_p})_{ci}(\delta T_{fi} - \delta T_{c_{2i}}) - \frac{2}{\tau}(\delta T_{c_{1i}} - \delta T_{c_{2i}}) \tag{A.5}$$

The Pressurizer Dynamics

The model of the pressurizer was developed based on mass, energy, and volume balance, and is given as

$$\frac{d\delta P_p}{dt} = B_1 \delta P_p + B_2 \delta W_w + B_3 \delta q \tag{A.6}$$

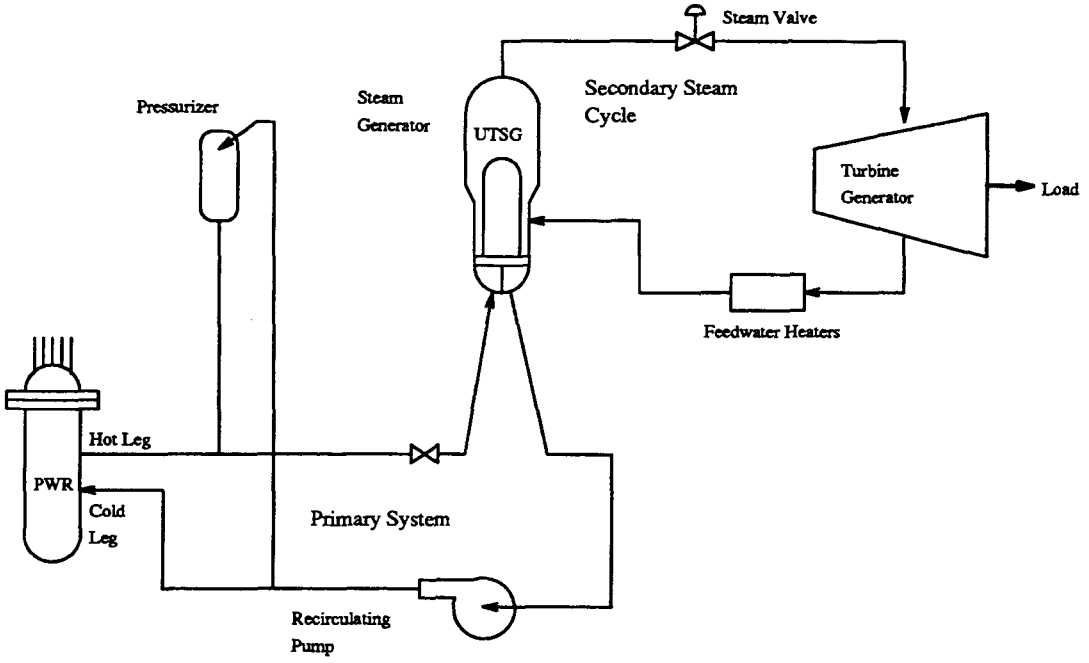


Figure A.1: PWR plant schematic

The Stream Generator

A simplified model that incorporates three regions–primary fluid, tube metal and secondary fluid–is used to represent the overall system. The equations that describe the steam generator subsystem are described as

$$\frac{d\delta T_p}{dt} = \frac{1}{\tau_{SG}}\delta T_{IP} - \frac{(hA_{pm})}{M_p C_p}(\delta T_p - \delta T_m) - \frac{1}{\tau_{SG}}\delta T_P \tag{A.7}$$

$$\frac{d\delta T_m}{dt} = \frac{(hA_{pm})}{M_m C_m}(\delta T_p - \delta T_m) - \frac{(hA_{ms})}{M_m C_m}(\delta T_m - \frac{\delta T_{sat}}{\delta P_s}\delta P_s) \tag{A.8}$$

$$\frac{d\delta P_s}{dt} = D_1\delta P_s + D_2\delta T_m + D_3\delta T_{FW} + D_4\delta W_{FW} + D_5\delta W_{so} \tag{A.9}$$

Plenums and Pipings

All the plenums and piping sections are modeled as well-mixed volumes described by

$$\frac{d\delta T}{dt} = \frac{1}{\tau}\delta T_{in} - \frac{1}{\tau}\delta T \tag{A.10}$$

The system distribution matrix A and control vector B are given as follows:

$$\mathbf{A} = \begin{bmatrix}
-400 & 0.077 & -1781 & -13700 & -13700 & 0 & 0 \\
400 & -0.077 & 0 & 0 & 0 & 0 & 0 \\
0.0756 & 0 & -0.16466 & 0.16466 & 0 & 0 & 0 \\
0 & 0 & 0.05707 & -2.4403 & 0 & 0 & 0 \\
0 & 0 & 0.05707 & 2.3262 & -2.3832 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.7664 & 0.5382 \\
0 & 0 & 0 & 0 & 0 & 3.0702 & -5.3657 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.349 \\
0 & 0 & 0 & 0 & 0.33645 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.45 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2.3832 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.2238 & 0 & 0 & 0 \\
0.3327 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.02034 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.33645 & 0 & 0 & 0 & 0 & 0 \\
0 & 2.5 & -2.5 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.45 & -1.45 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1.45 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.48 & 1.48 \\
0 & 0 & 0 & 0 & 0 & 0.516 & -0.516
\end{bmatrix}$$

$$\mathbf{B^T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.04425 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A.2 Plant Data

General:	
Reactor type	Slightly enriched uranium, light water moderated and cooled pressurized water reactor
Number of reactors in plant	1
Number of steam generators and type	3 vertical U-tube recirculation type steam generators
Number of pressurizers	1
Rated output per reactor	2200 MW (th), 739 MW (e)
Net efficiency	33.6%
Location	Raleigh, North Carolina, USA

Core Thermal and Hydraulic Characteristics:	
Total primary heat output	2200 MW (th)
Nominal primary system pressure $\mathbf{P_o}$	2250 p.s.i.
Total coolant flow rate	101.5e6 lb/hr
Average coolant velocity	14.3 ft/sec.
Total mass of coolant in primary loop $\mathbf{M_1}$	406050 lb
Nominal coolant inlet temperature $\mathbf{T_{c_2}}$	546.2 F
Nominal coolant outlet temperature $\mathbf{T_{c_1}}$	602.1 F
Active heat transfer surface area $\mathbf{A_f}$	42460 ft^2
Average heat flux	171600 Btu(hr.ft^2 F)
Fuel to coolant heat transfer coefficient $\mathbf{U}$	176 Btu/ (hr.ft^2 F)

Kinetic Characteristics:	
Doppler coefficient α_f	-1.3e-5 $(\Delta k/k)F$
Moderator temperature coefficient α_c	-2.0e-4 $(\Delta k/k)F$
Prompt neutron life time	1.6e-5 sec.
Delayed neutron fraction β	0.0064

Delayed neutron constants :		
Mean life (sec)	Decay constant (λ_i sec)	Fraction
80.4	0.0124	0.00021
32.8	0.0305	0.00140
8.98	0.111	0.00125
3.32	0.301	0.00253
0.88	1.14	0.00074
0.332	3.01	0.00027

Steam Generator:	
Number of U-tubes	3260
U-tube diameter	0.875 in.
Average tube wall thickness	0.05 in
Mass of U-tube metal $\mathbf{M_m}$	91800 lb
Total heat transfer area $\mathbf{A}$	44430 ft^2

Steam Condition at Full Load :	
Steam flow $\mathbf{W_{so}}$	3.169e6 lb/hr
Steam temperature $\mathbf{T_s}$	516 F
Steam pressure $\mathbf{P_s}$	770 psig

Primary Side Coolant:	
Reactor Coolant flow $\mathbf{W_{w_1}}$	33.93e6 lb/hr
Reactor Coolant Water Volume $\mathbf{V_{w_1}}$	928 ft^3

Secondary Side Fluid:	
Feed water temperature $\mathbf{T_{1FW}}$	435 F
Secondary side water volume $\mathbf{V_{w_2}}$	1526 ft^3
Secondary side steam volume $\mathbf{V_{s_2}}$	3203 ft^3

Pressurizer Design Data :	
Water volume $\mathbf{V_w}$	780 ft^3
Steam volume $\mathbf{V_s}$	520 ft^3
Electric heater capacity	1300 kw

APPENDIX B

THE BACKPROPAGATION LEARNING ALGORITHM

B.1 The Generalized Delta Rule

The backpropagation algorithm is a conceptually generalized version of the least-mean-squares algorithms. The former algorithm adopts a learning rule called the Generalized Delta Rule (GDR). GDR uses a gradient descent technique to minimize an objective function equal to the sum of the mean square error E between the desired and the actual network outputs.

The error function can be written as:

$$E = \sum_{p=1}^{P} E_p \tag{B.1}$$

where P is the number of training patterns, E_p is the total squared error for the pth

pattern:

$$E_p = 1/2 \sum_{j=1}^{M} [t_{pj} - O_{pj}]^2 \tag{B.2}$$

where M is the number of nodes in the output layer, t_{pj} is defined as the desired value of the jth output element given the pth pattern, and O_{pj} is the actual output of the same element.

Given the pth pattern, the network uses the input vectors and produces the actual output; if there is no difference between those two there will be no learning otherwise the weights will be changed by

$$\Delta_p w_{ji} = \alpha(t_{pj} - O_{pj}) I_{pi} = \alpha \delta_{pj} I_{pi} \tag{B.3}$$

where I_{pi} denotes the value of ith element of the input pattern p and w_{ji} are the connection weights. Let

$$\delta_{pj} = t_{pj} - O_{pj} \tag{B.4}$$

$\Delta_p w_{ji}$ is the change to be made to the weight from ith to jth pattern p.

Delta rule implements a gradient descent in E_p when units are linear.

$$-\frac{\delta E_p}{\delta w_{ji}} = \delta_{pj} I_{pi}$$

$$\frac{\delta E_p}{\delta w_{ji}} = \frac{\delta E_p}{\delta O_{pj}} \frac{\delta O_{pj}}{\delta w_{ji}} \tag{B.5}$$

The first part of this rule describes how the error changes with the output of the jth unit and the second part describes how much change in w_{ji} changes that output. In order to get correct generalization of the delta rule, we must set

$$\Delta_p w_{ji} \propto -\frac{\delta E_p}{\delta w_{ji}} \tag{B.6}$$

Let us define

$$\delta_{pj} = -\frac{\delta E_p}{\delta a_{pj}} \tag{B.7}$$

this is consistent with the definition of δ_{pj} used in the original delta rule for linear units since $O_{pj} = a_{pj}$ when the unit U_j is linear; where $a_{pj} = \sum_i w_{ij} O_{pi}$. The equation has the form

$$-\frac{\delta E_p}{\delta w_{ji}} = \delta_{pj} O_{pi} \tag{B.8}$$

To implement gradient descent in E_p we should make the weight changes according to

$$\Delta_p w_{ji} = \alpha \delta_{pj} O_{pi} \tag{B.9}$$

To compute $\delta_{pj} = -\frac{\delta E_p}{\delta a_{pj}}$, we apply the chain rule

$$\delta_{pj} = -\frac{\delta E_p}{\delta a_{pj}} = -\frac{\delta E_p}{\delta O_{pj}} \frac{\delta O_{pj}}{\delta a_{pj}} \tag{B.10}$$

First part of this derivation reflects the change in error as a function of the output of the unit, the second reflects the change in the output as a function of changes in the input. The second factor can be computed by

$$\frac{\delta O_{pj}}{\delta a_{pj}} = f_j'(a_{pj}) \tag{B.11}$$

Therefore, the error signal δ_{pj} in eqn B.4 is computed as follows:

$$\delta_{pj} = \begin{cases} [t_{pi} - O_{pi}] f_j'(a_{pj}) & \text{if } j \in \text{the output layer} \\ f_j'(a_{pj}) \sum_k \delta_{pk} w_{pkj} & \text{if } j \in \text{the hidden layer(s)} \end{cases} \tag{B.12}$$

where f_j' is the derivative of the sigmoid function of the jth node. f' depends on the activating function. For a sigmoid function

$$\frac{\delta O_{pj}}{\delta a_{pj}} = O_{pj}(1 - O_{pj}) \tag{B.13}$$

The learning procedure requires that the change in weight be proportional to $\frac{\delta E_p}{\delta w_{ij}}$. The constant of proportionality is the learning rate of this procedure. Large, constant learning rates, may lead to oscillation. In order to prevent this oscillation another term has been introduced and is called *the momentum term.* The new weight updating formula has the form

$$w_{ji}(n+1) = \eta(\delta_{pj}O_{pj}) + \alpha w_{ji}(n) \tag{B.14}$$

where η is the learning rate, α is the momentum term which determines the effect of past changes on the current direction of movement in weight space. This term provides a kind of momentum in weight space that effectively filters out high frequency variations of the error surface in the weight space.

In summary, the major goal of the learning algorithm is for the activations on the output layer after proceeding in feedforward manner to match those in the training data target. The error signal concentrates on changing those weights most influent on the output error. This algorithm calculates an error for each node in the hidden and output layers using eqn B.12, and recursively updates the connection strengths of all layers using eqn B.14, beginning from the output layer and propagating backward until the input layer as shown in Figure B.1.

B.2 The Backpropagation Network Algorithm

The backpropagation network algorithm uses the Generalized Delta Rule for training. The algorithm can be summarized to the following steps:

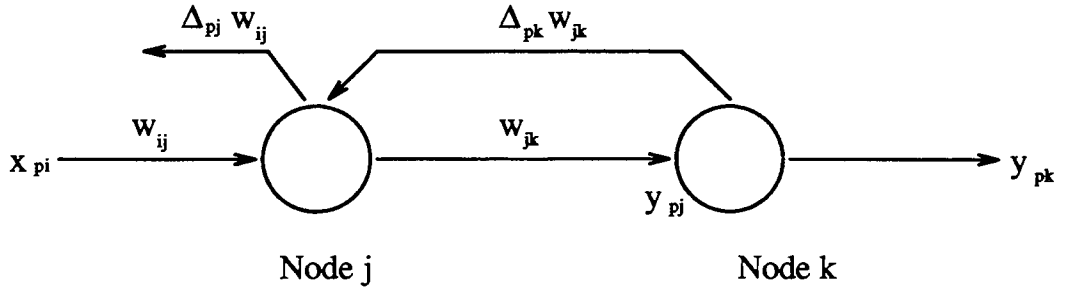


Figure B.1: Backpropagation of the error signal

1. Initialize the weights w_{pij} and bias θ_{pj} to all nodes using random numbers in the range [-1,1].

2. Present the input vector to the first layer of the network, and compute the output vector by using the following equation

$$x_{pj} = \frac{1}{1 + e^{-(\sum_i y_{pi} w_{pij} + \theta_{pj})}}$$

3. Compute the local error between the desired value and the actual value for each node at the output layer as

$$\delta_{pj} = x_{pj}(1 - x_{pj})(t_{pj} - x_{pj})$$

4. Calculate the local error for each node in the hidden layers by using

$$\delta_{pj} = x_{pj}(1 - x_{pj})(\sum_i \delta_i w_{(p+1)ij})$$

5. Compute all the weight adjustments as

$$\Delta_p w_{ij} = \alpha \delta pj x_{pi}$$

and the bias adjustments as

$$\Delta\theta_{pj} = \alpha\delta_{pj}$$

6. Update all the connection weights by adding the weight adjustments to the old weights as

$$w_{pij}(n+1) = w_{pij}(n) + \Delta_p w_{ij}(n+1)$$

7. Update all the bias values by adding the bias adjustments to the previous bias values by using

$$\theta_{pj}(n+1) = \theta_{pj}(n) + \Delta\theta_{pj}(n+1)$$

8. Repeat (step 2 through 7) until the error E_p is sufficiently small.

REFERENCES

[1] J. A. Anderson, **'A simple neural network generating an interactive memory,'** *Mathematical Biosciences*, Vol. 14, pp.197-220 , 1972.

[2] G. Basile and G. Marro, **'On the observerbility of linear time-invariant systems with unknown inputs,'** *Journal of Optimization Theory and Application*, Vol.3, No. 6, pp.410-415 , 1969.

[3] M. Basseville, **'Detecting changes in signals and systems–A survey,'** *Automatica*, Vol.24, pp.309-326, 1988.

[4] R.D. Beard, **'Failure accommodation in linear systems through self-recognition,'** *Rept. MVT-71-7*, Man-Vehicle Laboratory, Cambridge, MA, 1971.

[5] C.T. Chen, **'Linear System theory and design,'** HRW Publishing, NY, 1984.

[6] R.N. Clark, **'Instrument fault detection,'** *IEEE Trans. Aero. Electron. Systems*, AES-14, pp.456-465 1978.

[7] R.N. Clark, D.C. Fosth and V.W. Walton **'Detecting instrument malfunctions in control systems,'** *IEEE Trans. Aero. Electron. Systems*, AES-11, pp465-473, 1975.

[8] J.C. Deckert, M.N. Desai, J.J. Deyst and A.S. Willsky, **'F-8 DFBW sensor failure identification using analytical redundancy,'** *IEEE Trans. Automatic Control*, Vol. AC-22, pp.795-803, 1977.

[9] E. Eryurek and B.R. Upadhyaya, **'Sensor validation for power plants using adaptive backpropagation neural network,'** *IEEE Nuclear Sciences Symposium*, San Francisco, January 1990.

[10] P.M. Frank, **'Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy–A survey and some new results,'** *Automatica*, Vol.26, pp459-474, 1990.

[11] P.M. Frank and L. Keller, **'Sensitivity discriminating observer design for instrument failure detection,'** *IEEE Trans. Aero. Electron. Systems*, AES-16, pp.460-467, 1980.

[12] Y. Guan and M. Saif, **'A novel approach to the design of unknown input observers,'** *IEEE Trans. Auto. Cont.*, M.A.Sc Thesis, Simon Fraser University, Burnaby, B.C., Canada, 1990.

[13] Y. Guan, **'Robust estimation with application to failure detection and identification (FDI) in dynamical systems**, Vol.36, pp.632-635, 1991.

[14] D.A. Handelman and R.F. Stengel, **'Combining expert system and analytical redundancy concepts for fault-tolerant flight control,'** *Journal of Guidance, Control, and Dynamics*, Vol.12, No.1, pp.39-45, Jan-Feb. 1989.

[15] D. Hebb, *The Organization of Behavior*, New York: Wiley , 1949.

[16] R. Hecht-Nielsen, **'Counter-propagation networks,'** *Proc. of IEEE First Inter. Conf. on Neural Networks*, Vol. II , pp.19-32, San Diego, CA. 1987.

[17] G. E. Hinton and T. J. Sejnowski, **'A learning algorithm for Boltzmann machines,'** *Cognitive Science*, Vol. 9, pp.147-169, 1985.

[18] J. J. Hopfield, **'Neural network and physical systems with emergent collective computational abilities,'** *Proc. Natl. Acad., Sci. U.S.A.,*, Vol.79 , pp.2554-2558 , April 1982.

[19] W.E. Hopkins, J. Medanic, and W.R. Perkins, **'Output feedback pole placement in the design of suboptimal linear quadratic regulators,'** *Int. J. Cont.* , Vol. 34, pp.593-612, March 1981.

[20] J.C. Hoskins and D.M. Himmelblau, **'Artificial neural network models of knowledge representation in chemical engineering,'** *Comput. Chem. Eng.*, Vol.12, pp881-890, 1988.

[21] C.Y. Huang and R.F. Stengel, **'Failure model determination in a knowledge-based control system,'** *Proceedings of the 1987 American Control Conference*, pp.1643-1648, Minneapolis, June 1987.

[22] R. Isermann, **'Process fault detection based on modelling and estimation methods–A survey,'** *Automatica*, Vol.20, pp.387-404, 1984.

[23] H.L. Jones, **'Failure detection in linear systems'**, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, Mass., September, 1973.

[24] T.W. Kerlin, **'Dynamic analysis and control of pressurized water reactors,'** in *Control and Dynamic systems–Advances in Theory and applications*, Vol. 14, C.T. Leondes, Ed., NY: Academic, pp.103-212, 1978.

[25] T.W. Kerlin, E.M. Katz, J.G. Thakkar and J.E. Strange , **'Theoretical and experimental dynamic analysis of the H. B. Robinson nuclear plant,'** *Nuclear Technology*, Vol.30, pp.299-316, Sept. 1976.

[26] I.S. Kim and M. Modarres, **'Application of goal tree-success tree model as the knowledge-base of operator advisory systems,'** *Nuclear Engineering and Design*, Vol. 104, pp.67-81, 1987.

[27] A. H. Klopf, **'A drive-reinforcement model of single neuron function: an alternative to the Hebbian neural model,'** *Proc. of Conf. on Neural Networks for Computing*, pp.265-270, Snowbird, UT, April 1986.

[28] T. Kohonen, *Self-Organization and Associative Memory*, 2nd Ed., New York: Springer-Verlag, 1987.

[29] B. Kosko, **'Adaptive bidirectional associative memories,'** *Applied Optics*, Vol. 36, pp.4947-4960, December 1987.

[30] M.A. Kramer and B.L. Palowitch, **'Expert system and knowledge-based approaches to process malfunction diagnosis,'** presented at the *AIChE* Nat. Meet., Chicago , 1985.

[31] J.E. Kurek, **'The state vector reconstruction for linear systems with unknown inputs,'** *IEEE Trans. on Auto. Cont.*, Vol.AC-28, pp.1120-1122, 1983.

[32] W. S. McCulloch, W. Pitts, **'A logical calculus of the idea immanent in neural nets,'** *Bulletin of Mathematical Biophysics*, Vol.5, pp.115-133. 1943.

[33] R.K. Mehra and J. Peschon **'An innovations approach to fault detection and diagnosis in dynamic systems,'** *Automatica* , Sept. 1971.

[34] W.C. Merrill, **'Sensor failure detection for jet engine using analytical redundancy,'** *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 8, pp.673-682, 1985.

[35] S.R. Naidu, E. Zafiriou and T.J. McAvoy, **'Use of neural networks for sensor failure detection in a control system,'** *IEEE Control System Magazine*, Vol.10, pp.49-55, April 1990.

[36] P.P. Nair, and M. Gopal, **'Sensitivity-reduced design for a nuclear pressurized water reactor,'** *IEEE Transactions on Nuclear Science*, Vol.34, N0. 6, pp.1834-1842, December 1987.

[37] K.S. Narendra and K. Parthasarathy, **'Identification and control of dynamical systems using neural networks,'** *IEEE Transactions on Neural Networks*, Vol.1, No.1, pp.4-27, March 1990.

[38] S.H. Rich and V. Venkatasubramanian, **'Model-based reasoning in diagnostic expert systems for chemical process plants,'** *Comput. Chem. Engng.*, Vol.11, pp.111-122 , 1987.

[39] F. Rosenblatt, **'Principles of neurodynamics: perceptrons and the theory of brain mechanisms,'** Spartan Books, Washington D.C., 1961.

[40] D.E. Rumelhart and J.L. McClelland, Eds., **'Parallel Distributed Processing,'** The MIT Press, Cambridge, MA, 1986.

[41] M. Saif, **'A novel approach for optimal control of a pressurized water reactor,'** *IEEE Transactions on Nuclear Science*, Vol.36, pp.1317-1325, Feb. 1989.

[42] M. Saif, **'Suboptimal projective control of a pressurized water reactor,'** *IEEE Transactions on Nuclear Science*, Vol.36, No.6, pp.2459-2465, December 1989.

[43] M. Saif and Y. Guan, **'Decentralized state estimation in large-scale interconnected dynamical systems,'** *Automatica*, Vol.28, No.1, pp.215-219, 1992.

[44] M. Saif and F.E. Villascca, **'Optimal control under distributed sensor failure detectors,'** *Control of Distributed Parameter Systems*, H.E.Rauch, Ed., Pergamon Press, pp.407-412, 1987.

[45] M. Saif and F.E. Villascca, **'Sensor failure detection in optimally controlled systems,'** *Proceedings of the ACC*, pp.2104-2110, Seattle, WA, 1986.

[46] R.F. Stengel, **'Intelligent failure tolerant control,'** *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, pp.548-557, Philadelphia, 1990.

[47] B.R. Upadhyaya, E. Eryurek and G. Mathai, **'Application of neural computing for signal validation,'** *Seventh Power Plant Dynamics, Control and Testing Symposium*, Vol.1, pp.27.01-27.18, May 1989.

[48] B.R. Upadhyaya, G. Mathai and E. Eryurek, **'Development and application of neural network algorithms for process diagnostics,'** *Proceedings of the 29th Conference on Decision and Control*, Honolulu, December 1990.

[49] V. Venkatasubramanian, and K. Chan, **'A neural networks methodology for process fault diagnosis,'** *AIChE J.*, Vol.35, pp.1993-2002, 1989.

[50] K. Watanabe, and D.M. Himmelblau, **'Fault diagnosis in nonlinear chemical processes:II. Application to a chemical reactor,'** *AIChE J.*, Vol.29, pp.250-258, 1983(b).

[51] K. Watanabe, I. Matsuura, M. Abe, M. Kubota and D.M. Himmelblau, **'Incipient fault diagnosis of chemical processes via artificial neural networks,'** *AIChE J.*, Vol.35, pp.1803-1812, 1989.

[52] A.S. Willsky, **'A survey of design methods for failure detection in dynamic systems,'** *Automatica*, Vol. 12, pp.601-611 , 1976.

[53] *Matlab^{TM} for Sun: User's Guide*, The Math Works Inc., S. Natick, MA, 1989.