To the Graduate Council:

I am submitting herewith a thesis written by Kadir Kavaklioglu entitled "Performance Monitoring of Pressurized Water Reactors Using Artificial Neural Networks." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R Upadhyaya

Belle R. Upadhyaya, Major Professor

We have read this thesis
and recommend its acceptance:

Robert E Uhrig

Lawrence F Miller

Accepted for the council:

Lew Winkel

Associate Vice Chancellor
and Dean of The Graduate School

PERFORMANCE MONITORING OF PRESSURIZED WATER REACTORS USING ARTIFICIAL NEURAL NETWORKS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee

Kadir Kavaklioglu

May 1993

DEDICATION

This thesis is dedicated to my parents

Mr. Halil Kavaklioglu

and

Mrs. Meryem Kavaklioglu

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Belle R. Upadhyaya, for his invaluable guidance and suggestions. I would like to thank the other committee members, Dr. Robert E. Uhrig, and Dr. Laurence F. Miller, and the department head Dr. Thomas W. Kerlin, for their comments and assistance. I would also like to thank my project teammates, Ali S. Erbay and Evren Eryurek, for their cooperation. I would like to express my thanks to my wife, Gul, for her understanding and patience.

This research is sponsored by the Tennessee Valley Authority under grant TVA TV-84395V. This assistance is gratefully acknowledged.

ABSTRACT

Performance monitoring of a Pressurized Water Reactor (PWR) plant, in general, includes the monitoring and diagnostics of plant systems (both primary and secondary), actuators and instrument channels. The primary goal of operation of a PWR is to maintain as high a thermal power as possible, to achieve a low heat rate (Btu thermal power / kWh electricity generated), and maintain safe operation. The focus of the present research is to apply the technology of Artificial Neural Networks (ANN) to the problems of estimating steam generator feedwater flow rate, plant heat rate, unaccounted heat losses, and efficiency of isolated balance-of-plant (BOP) systems. All of these are important aspects of achieving a high degree of performance of a power plant. The necessity of developing an independent approach for estimating feedwater flow rate has been motivated by the observed error in its measurement, possibly due to the venturi fouling phenomenon.

Multi-layer perceptron (MLP) networks that use the backpropagation training algorithm were applied to seek solutions to the above problems. Both heteroassociative and autoassociative networks were developed using a PC-based software system. Operational data were obtained from two different PWR plants. Data preprocessing was necessary because of the varying magnitudes and the dimensionality of input and output information used for the networks. Training patterns (plant measurements) were selected based on the physical principles governing the process. An output sensitivity analysis of the network was performed to identify the important variables influencing the feedwater flow rate. This is a statistical approach applied to a trained network. In some cases, the estimations from neural networks were compared with calculations based on thermodynamic

principles. The size of the hidden layer of a neural network was determined by a trial-and-error approach.

Both neural network and phenomenological models were applied to the estimation of performance related parameters of PWRs. Data from a four-loop PWR (United States), and a six-loop PWR (Europe) were used in this research task. The results indicated the feasibility of applying the methodology developed in this research. The success of this advanced technology is very much dependent on the availability of reliable measurements and a properly generated data base. The plant performance monitoring is part of an overall Fault-Tolerant Control and Diagnostics System (FCDS) being developed at The University of Tennessee.

TABLE OF CONTENTS

CHAPTER PAGE

1 INTRODUCTION 1

1.1 Statement of the Problem 1

1.2 Review of Prior Work 4

1.3 Overview of the Proposed Methodology 5

1.4 Contributions of the Thesis 6

1.5 Outline of the Thesis 8

2 NEURAL NETWORKS AND POWER PLANT ANALYSIS 10

2.1 Introduction 10

2.2 Thermal Performance Monitoring 11

2.2.1 Process Pattern Monitoring Using Autoassociative Networks 11

2.2.2 Heat Loss Monitoring 12

2.2.3 Heat Rate Monitoring 13

2.2.4 Feedwater Flow Monitoring 19

2.2.4.1 The Venturi Meter 24

2.2.4.2 Venturi Fouling 26

2.2.4.3 Neural Network Models 27

2.3 Thermal Performance Diagnostics 27

2.4 General Approach Using Neural Networks 31

3 ARTIFICIAL NEURAL NETWORKS 32

3.1 Introduction 32

3.2 Artificial Neural Networks 34
3.3 Backpropagation Network Algorithm 37
3.3.1 General Properties 37
3.3.2 The Recall Process 39
3.3.3 The Training Process 39
3.3.4 The Training Algorithm 40
3.4 Traditional versus Neural Network Models 43
4 NETWORK PARAMETERS AND DATA PREPARATION 45
1.1 Introduction 45
4.2 Selection of Input Signals 45
4.3 Data Preprocessing 47
4.4 Training Data 48
4.5 Steady-State versus Dynamic Data 48
4.6 Number of Hidden Layers 49
4.7 Number of Hidden Layer Nodes 50
4.8 Other Parameters 50
5 NETWORK SENSITIVITY ANALYSIS 52
5.1 Introduction 52
5.2 Differential Sensitivity Measure 52
5.3 Statistical Sensitivity Measure 55
6 RESULTS AND DISCUSSION 59
6.1 Introduction 59
6.2 Process Pattern Monitoring Using Autoassociative Networks 60
6.3 Heat Loss Estimation 72
6.4 Heat Rate Estimation 74

6.5 Feedwater Flow Estimation 83
6.5.1 A Four-loop Pressurized Water Reactor 83
6.5.2 A Six-Loop Pressurized Water Reactor 88
6.5.2.1 Least-Squares Estimation of Flow Rate 90
6.5.2.2 Neural Network Estimation 91
6.6 Sensitivity Analysis 95
6.7 Thermal Performance Diagnostics 97
6.8 On-line Implementation 100
7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH 105
7.1 Summary 105
7.2 Conclusions 107
7.3 Recommendations for Future Work 110
LIST OF REFERENCES 112
APPENDICES 117
Appendix A 118
Appendix B 125
Appendix C 137
VITA 139

LIST OF TABLES

TABLE PAGE

6.1 List of Signals for Auto-Associative Monitoring 62

6.2 Learning and Recall Root-Mean-Square Errors (RMSE) 71

6.3 List of Signals Used for Heat Loss Estimation 73

6.4 Learning Root-Mean-Square Errors 79

6.5 Recall Root-Mean-Square Errors 79

6.6 List of Signals for Heat Rate Estimation 81

6.7 List of Signals Used for Feedwater Flow Analysis of Four-Loop PWR 84

6.8 Learning and Recall RMSEs for Feedwater Estimations 85

6.9 List of Signals Used for Feedwater Flow Analysis of a Six-Loop PWR 89

6.10 Learning (L) and Recall (R) RMSEs for Neural Network (NN) and Least-Squares (LS) Estimations 92

6.11 Order of Importance of Inputs for Feedwater Flow Rate 99

6.12 List of Signals Used for Diagnostics 101

LIST OF FIGURES

FIGURE PAGE

2.1 Secondary Side of a Typical PWR Plant 15

2.2 Global Schematic of the Secondary Side 17

2.3 Schematic of a Steam Generator 20

2.4 Schematic of a Venturi Meter 22

2.5 Schematic of a Typical Feedwater Heater 29

3.1 Schematic of a Neuron 33

3.2 Schematic of a Typical Processing Element 35

3.3 General Topology of a Multi-Layer Perceptron Network 38

3.4 The Sigmoid Function 41

5.1 A Three-Layer Single-Output Neural Network 54

6.1 Measured versus Estimated Main Steam Pressures in MSR A1 63

6.2 Measured versus Estimated Impulse Pressures 64

6.3 Measured versus Estimated CCW Inlet Temperatures 65

6.4 Measured versus Estimated Steam Throttle Pressures 66

6.5 Measured versus Estimated RCS Pressures 67

6.6 Measured versus Estimated Cold Leg Temperatures 68

6.7 Measured versus Estimated Hot Leg Temperatures 69

6.8 Measured versus Estimated Feedwater Pressures 70

6.9 Calculated versus Estimated Heat Losses for Learning for Unit-1 75

6.10 Calculated versus Estimated Heat Losses for Recall for Unit-1 76

6.11 Calculated versus Estimated Heat Losses for Learning for Unit-2 77

6.12 Calculated versus Estimated Heat Losses for Recall for Unit-2 78
6.13 Calculated versus Estimated Gross Heat Rates 82
6.14 Measured versus Estimated Feedwater Flow Rates for Unit-1 86
6.15 Measured versus Estimated Feedwater Flow Rates for Unit-2 87
6.16 Measured (M) versus Neural Network (NN) and Least-Squares (LS) Estimated Feedwater Flow Rates for Loop-1 93
6.17 Measured (M) versus Neural Network (NN) and Least-Squares (LS) Estimated Feedwater Flow Rates for Loop-2 94
6.18 Differential Sensitivity Results 96
6.19 Statistical Sensitivity Results 98
6.20 Calculated versus Estimated Feedwater Heater Efficiencies 102
6.21 Main Menu of FCDS 103
6.22 Performance Monitoring Screen of FCDS 104

LIST OF ACRONYMS

ANN	:	Artificial neural network
BOP	:	Balance-of-plant
BP	:	Backpropagation
C	:	Condenser
CCW	:	Condenser cooling water
CL	:	Cold leg
FCDS	:	Fault-Tolerant Control and Diagnostics System
FWH	:	Feedwater heater
FWP	:	Feedwater pump
G	:	Generator
GHR	:	Gross heat rate
HDP	:	Heater drain pump
HL	:	Hot leg
HLE	:	Hidden layer elements
HLPE	:	Hidden layer processing elements
HPT	:	High pressure turbine
HR	:	Heat rate
L	:	Learning
LPT	:	Low pressure turbine
LS	:	Least-squares
M	:	Measurement
MFWP	:	Main feedwater pump

MLP	:	Multi-layer perceptron
MSR	:	Moisture separator reheater
NN	:	Neural network
NRC	:	Nuclear Regulatory Commission
PWR	:	Pressurized water reactor
R	:	Recall
RCS	:	Reactor coolant system
RMSE	:	Root-mean-square error
SG	:	Steam generator

Chapter 1

INTRODUCTION

1.1 Statement of the Problem

One of the primary objectives of the power plant industry has long been the efficient operation of plant systems, thus reducing the cost of electricity. In general, the measure of efficiency is the plant heat rate, which is the amount of heat supplied to the system to produce one kilowatt-hour electrical energy. Especially for nuclear power generating stations, there is another equally important objective. This is to operate the plant at full rated power.

Efficient operation of a power plant requires the monitoring of all critical process variables [1]. These are the variables or parameters which contribute significantly to thermal power or heat rate. However, there are too many components and subsystems which are highly coupled in the whole system. This complexity makes the monitoring of every single component or signal by classical analytical or empirical methods a voluminous task.

The operating power of a nuclear reactor is limited by the Nuclear Regulatory Commission (NRC) licensing requirements. Therefore, the thermal power of the plant should be determined very accurately. Currently, classical thermodynamic methods utilizing mass and energy balances on steam generators are used for thermal power calculation.

The feedwater flow rate to the steam generator is one of the major quantities used in determining thermal power. This flow rate is measured by installed venturi tubes. Pressure drop across the venturi tube (by using pressure taps at the inlet and the throat in the piping) is measured and is related to the flow rate in the piping [2] [3].

Although the use of factors such as thermal expansion factor or adiabatic expansion factor in the pressure drop and mass flow rate conversion equation may introduce some errors, they can be eliminated by proper techniques such as sensor calibration. However, the most important problem in feedwater flow determination is the deposition of corrosion products on the venturi converging section and throat over a period of time. As a result, the measured flow rates are higher than their actual values, thus yielding artificially high plant thermal powers, and unwarranted plant power deratings [4]. A 2% power derating would cost about $20,000 per day in lost revenue for an 800 MWe unit at a utility rate of $0.05/kWh [5]. Since venturi meters are used in both PWRs and BWRs, the total lost revenues could be very large. An artificial neural network trained with reliable data can estimate the correct flow rate (used in thermal power calculations) regardless of the changes in the measurement system or the physical state of the feedwater [6].

Plant heat rate depends on almost all process variables of the plant. Calculation of heat rate is based on thermodynamic equilibrium and assumptions of ideal processes. It requires the use of steam property tables for equilibrium properties and some empirical data (in the form of look-up tables or graphs) for non-idealistic corrections. It is desirable to operate the plant as efficiently as possible. Therefore, it is necessary to maintain high

efficiencies of all components and to minimize the heat losses. Properly developed neural networks may be used to estimate the heat rate of the plant accurately.

In addition, when there is a change in the performance of the plant, it is desirable to determine the cause and make corrections immediately. Neural networks may also be used for diagnostics of the performance of the plant. A neural network can be trained for each component of the plant whose single output is the efficiency of that specific component. Then the outputs of these networks can be treated as the inputs to another network whose output is the efficiency of the overall system.

Backpropagation network algorithm is one of the most widely used paradigms. This learning algorithm has proven its ability to make a layered network represent mappings and has been applied to many problems [7] [8]. However, there are certain issues to be addressed in applying this technology to specific problems. These include determining the number of hidden layers, the number of hidden layer processing elements, data preprocessing, learning coefficients, etc. The aim is to reduce the estimation error and the training time.

This thesis addresses the following issues as they are applied to research and development in performance monitoring of PWR systems.

1. Selection of input signals to neural networks and data preprocessing
2. Development of a heat loss estimation methodology
3. Development of a feedwater flow rate monitoring methodology
4. Sensitivity analysis to identify the effect of input variables on the network output
5. Development of a heat rate monitoring methodology

6. Development of a sub-system level thermal performance diagnostics technique
7. Comparison with existing approaches.

1.2 Review of Prior Work

Most of the studies performed by the utilities or the vendors, involving the determination of correct feedwater flow rates are mainly re-calibration of venturi meters or finding ways to correct the flow rate by modifying the conversion equations by some analytical means [9]. One very popular method of correcting the flow rate is to use the primary and secondary thermal power balances. On the other hand, there are some studies on feedwater flow estimation or system diagnostics which do not focus on venturi fouling problem [10] [11].

Current heat loss evaluation programs include the use of vendor supplied efficiency curves which give heat losses as a function of one or more process variables [10]. An overall power accounting using thermodynamic principles is also used for obtaining this loss. Using correctional methods is one of the remedies for reducing the unaccountable part of the heat loss [13].

There is a large amount of work in the field of monitoring heat rate of fossil and nuclear power plants [14] [15] [16]. These efforts have evolved from hand calculations using occasional test data, to off-line simulation, and finally to special purpose on-line calculations and data display. Recently, with the advances in hardware and software power in computation, integrated systems which satisfy all the needs of operation have

been available. However, there are a few cases in which Artificial Neural Network (ANN) methodology has been applied specifically to determine the heat rate [17].

One of the milestones for thermal performance diagnostics is a report titled "Thermal Performance Diagnostics Manual for Nuclear Power Plants" by Combustion Engineering prepared for Electric Power Research Institute [18]. Its purpose is to provide engineering personnel working at PWRs or BWRs with a consistent way of identifying thermal performance problems. Like the other studies in this field, it provides a rule base for an expert system to find the cause of degradation [19] [20].

1.3 Overview of the Proposed Methodology

This present study utilizes a multi-layer neural network backpropagation (BP) algorithm for monitoring and diagnostics of process variables in a system. Neural networks are intrinsically parallel and non-algorithmic methods. The ability of the BP algorithm to learn any arbitrary nonlinear mapping from inputs to outputs , and the fault-tolerant property of a multi-layer network have a variety of applications in power plant industry [21].

The mapping is performed using a multi-layer, fully connected, heteroassociative network. The network is said to be heteroassociative when the input and the output layers are not identical. The BP algorithm computes the weights between pairs of processing elements so that the difference between the actual output and the network output is minimized in a least-squares sense. The output layer, in general, consists of several elements, but if the goal is to predict one variable as a function of others, then the output

will be a single element. This algorithm can also be used for identical input and output sets for overall system monitoring purposes. In this case, the network is said to be auto-associative.

In general, the multi-layer perceptron (MLP) is capable of performing arbitrary mapping between inputs and the outputs without knowing the complete system specifications. Information about the system is incorporated into the network during the training phase. It is also possible for neural networks to update the models by renewing the connection weights.

For some of the tasks presented in this thesis, such as heat rate monitoring and feedwater flow rate estimation, physical models were also developed for the calculation of the performance related variables. These models are based on conservation of mass and energy. The outputs of these models were used to compare the neural network methodology with the classical methods.

1.4 Contributions of the Thesis

The major accomplishment of this research is the design and the implementation of a neural network methodology as both off-line and on-line performance monitoring and diagnostics tool with emphasis on the secondary side of a PWR plant. Another important accomplishment is that actual plant data from two PWRs were used in all the analyses. The following tasks were performed and they indicate specific contributions of this thesis.

One of the important issues affecting the performance of a neural network is the selection of training patterns. Since the neural network learns the relationship between the input and the output pairs of training set, the patterns which represent the correct (desired) relationship should be selected. For all the applications considered in this study, attention was paid to choose the patterns for which expected values match with the measurements, since these represent the correct system information.

Data pre-processing was found to be an important stage in neural network applications for avoiding network saturation and for accelerating the network training. Because of the largely varying magnitudes of the input and the output signals, a column-wise linear transformation technique was used to transform original values into the interval [0.1, 0.9] for some studies and to the interval [0.2, 0.8] for the other applications.

Before creating a neural network for a special application, various signals from the power plant were monitored using auto-associative networks to detect any anomaly or abnormal operating condition or change in the system behavior [22].

For the estimation studies (feedwater flow rate, plant heat rate, etc.) single output hetero-associative neural networks were used. Output processing element is the variable to be estimated and the input elements were determined based on the macroscopic mass and energy balances on the related components or the system.

An overall efficiency of a specific balance-of-plant (BOP) component was estimated as a means of monitoring its behavior. Incipient problems may be detected by this simple

application of neural network model. The fourth feedwater heater of a Westinghouse type PWR plant was considered for this analysis.

A sensitivity analysis was performed for the feedwater flow estimator networks in order to evaluate the important variables contributing to the flow rate. Relative and statistical sensitivities were used as sensitivity measures.

One other contribution of this thesis is the comparison of the neural network methodology developed here, with the classical methods currently in use in some PWRs. For some of the tasks presented in this thesis, such as heat rate and feedwater flow rate estimation, physical models were developed for the calculation of performance related variables. These models are based on thermodynamic principles.

Finally, two of the tasks, namely the feedwater flow estimation and the heat rate estimation parts of this thesis, were implemented in a special power plant fault-tolerant control and diagnostics system as an on-line monitoring tool.

1.5 Outline of the Thesis

In chapter two, various aspects of nuclear power plant analysis and how ANN are applied to these problems are discussed. The approach of estimating heat rate, heat losses, feedwater flow rate are presented in detail. The last part of this chapter explains how neural networks are applied to specific problems.

Chapter three outlines the backpropagation algorithm for adjusting the connection weights in a multi-layer network. After a brief introduction to neural networks, the procedures used for data handling, training and recall, and determining the key parameters of network models is the subject of chapter four.

A formulation of the differential and statistical measures for the output sensitivity of a neural network to any of its inputs is presented in chapter five. The results of application using operational PWR data are presented and discussed in chapter six. Conclusions and recommendations for future work are given in chapter seven.

Chapter 2

NEURAL NETWORKS AND POWER PLANT ANALYSIS

2.1 Introduction

In today's environment, licensing delays, the cost of funding, and escalating construction costs have made it difficult and expensive for utilities to add new power generating stations. The utilities, therefore, find it very important to improve the performance of their existing power plants to partly meet the power demand.

One key to improving the performance of a generating station is monitoring and correcting the component malfunctions and other problems. Even though these may not result in significant operational problems like plant shutdowns, may cause small decreases in plant efficiency over a period of time. A nuclear power plant is a complex energy conversion system of components and thermodynamic processes. Component malfunctions and process imbalances can result in reduced thermal efficiency and thermal power without causing significant plant operational problems. It is generally not easy to determine the existence of a problem, or the source of the problem, or the magnitude of the problem. There are two reasons for this.

1. The sheer size and the complexity of the system
2. The inter-relationships of the thermodynamic processes tend to mask the symptoms of many problems.

The task of identifying and quantifying these problems is the main concern of operators and performance engineers. Over a period of years the operators and engineers gain knowledge and experience and become adept at monitoring and diagnosing performance problems. However, the current calculational methods or measurement systems used by the plant staff may not give satisfactory results for some cases.

This thesis presents an approach for performance monitoring and diagnostics of PWRs using artificial neural networks. In this chapter, the specific tasks associated with the objectives of this research are discussed.

2.2 Thermal Performance Monitoring

Although the term thermal performance monitoring covers too many issues in power plant terminology, only the ones which are considered to be the most important are analyzed in this study.

2.2.1 Process Pattern Monitoring Using Autoassociative Networks

The secondary side of a nuclear power plant (the balance-of-plant) is a highly complex energy conversion system which utilizes water as working fluid, and converts heat into mechanical and then into electrical energy. This system has steam generators, turbines, reheaters, moisture separators, a condenser, feedwater heaters, pumps and necessary piping. The effective operation of this system requires monitoring of all the signals affecting the power level and heat rate. However, because of the large number of signals, monitoring them separately may become a formidable task.

Using neural networks in the auto-associative mode in determining the transient or abnormal conditions in nuclear power plants has been studied and found to be useful [23] [24]. This technique is based on the fact that the various signals related to a process form a pattern representing the current state of the plant. Normal operational states of a nuclear power plant can be coded into a neural network which later can be used to determine the abnormal state.

Artificial neural networks provide a means for real time pattern recognition which can be used in monitoring the overall process instead of monitoring the signals of the process separately. The variables of a system form a different pattern when there is a disturbance in the process. The detection of the change in the pattern, as the output of a neural network model, provides information about disturbance or deviation from the nominal state.

Multi-layer feed-forward networks were used for auto-associative monitoring of various signals of a process in a typical four loop PWR. In this application, there are several inputs and outputs, and the inputs and the outputs are the same. Two-layer (one hidden layer) networks were used for this purpose. The use of this continuous mapping not only provides a classification of the state of the system, but gives a functional relationship which can be used to quantify the changes in the process variables.

2.2.2 Heat Loss Monitoring

In the balance-of-plant system (secondary side), it is necessary to minimize the heat losses. This is the difference between the generator output power and the expected plant

electrical power calculated from the plant thermal power and efficiency. These losses result from errors in measurements and errors in thermal calculations. Some parts of these losses, such as due to the low pressure turbine efficiency, changes in the environmental conditions can be determined analytically using vendor provided efficiency curves and thermal calculations. The term "unaccounted heat loss" is used for the other portion of the total heat loss which cannot be evaluated by classical methods.

It is desirable to limit the so called unaccounted losses to less than 5 MWe power for a 1200 MWe power generating station. However, this loss may sometimes reach as much as 20 MWe which is highly undesirable. The causes of this loss may be the degradation of components, errors in the measurements or too much change from the nominal operational state in the plant. Accurate evaluation of this loss is very important in plant monitoring.

Artificial neural networks may be used to obtain this quantity based on the data from the calculations where the unaccounted loss is very small. Two-layer, single output networks were used to evaluate the unaccounted losses. Highly complex relationship between the unaccounted heat losses and the various plant variables may be represented by a neural network model.

2.2.3 Heat Rate Monitoring

Heat rate is a major factor in the overall performance of a power plant. Therefore, an accurate determination of plant heat rate, immediate detection of any degradation in heat rate, and identification of the sources are highly desirable.

Historically, most utilities monitored heat rate on a weekly basis using the ratio of the heat produced by the reactor to the average electrical power generated [19]. This practice is useful as a crude measure of the performance of the plant. But, it does not provide either operators or performance engineers with any useful on-line information pertinent to improving methods of plant operation throughout changing conditions.

Recently, some power plants have installed on-line performance monitors which provide instantaneous output on plant heat rate, as well as actual and target values of plant parameters [19]. These monitors can provide detailed numerical information, graphical comparison, and trend charts that indicate unit performance. In addition, combined with the expert systems, these programs can determine the root causes of the degradation in the heat rate.

There are various methods to calculate the heat rate of a power plant depending on the desired accuracy in this performance variable. The most theoretical method is to analyze the whole system (every flow path and component) thermodynamically (see Figure 2.1). For the sake of simplicity, a temperature-entropy diagram corresponding to the system can also be drawn. The heat transferred and the work done on each path of Figure 2.1 are calculated using the first law of thermodynamics and the steam tables, assuming that the processes are idealistic. Finally, these values for heat and work are used to evaluate the heat rate of the plant. Although this is a difficult method, it provides the information on the ideal performances of the sub-systems and flow paths besides the overall heat rate of the plant.

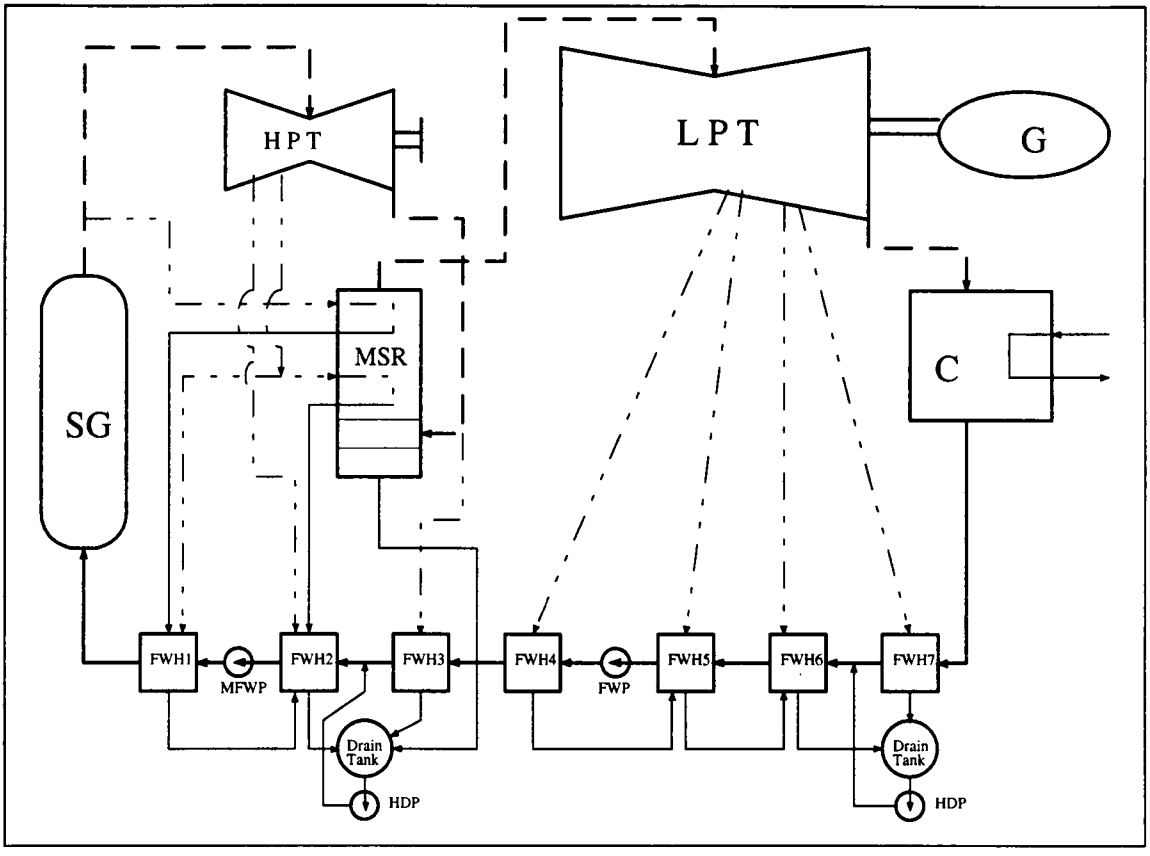


Figure 2.1: Secondary Side of a Typical PWR Plant.

Another approach for calculating the heat rate of a plant is to use the measured generator electrical power output. The plant can be modeled as a whole with the inputs of steam flow and vessel blowdown flow, and the outputs feedwater flow and generator output (see Figure 2.2). In this method, heat rate can be written as the combination of three quantities.

$$HR = \alpha + \beta - \gamma \qquad (2.1)$$

where

α : contribution of steam flow to heat rate

β : contribution of vessel blowdown flow to heat rate

γ : contribution of feedwater flow to heat rate

Steam flow contribution to heat rate is given by

$$\alpha = \frac{w^s(0.9975h_v^s - 0.0025h_l^s)}{P} \qquad (2.2)$$

where

w^s : mass flow rate of main steam

h_v^s : enthalpy of vapor phase of main steam

h_l^s : enthalpy of liquid phase of main steam

P : measured generator output

Vessel blowdown flow's contribution to heat rate is given by

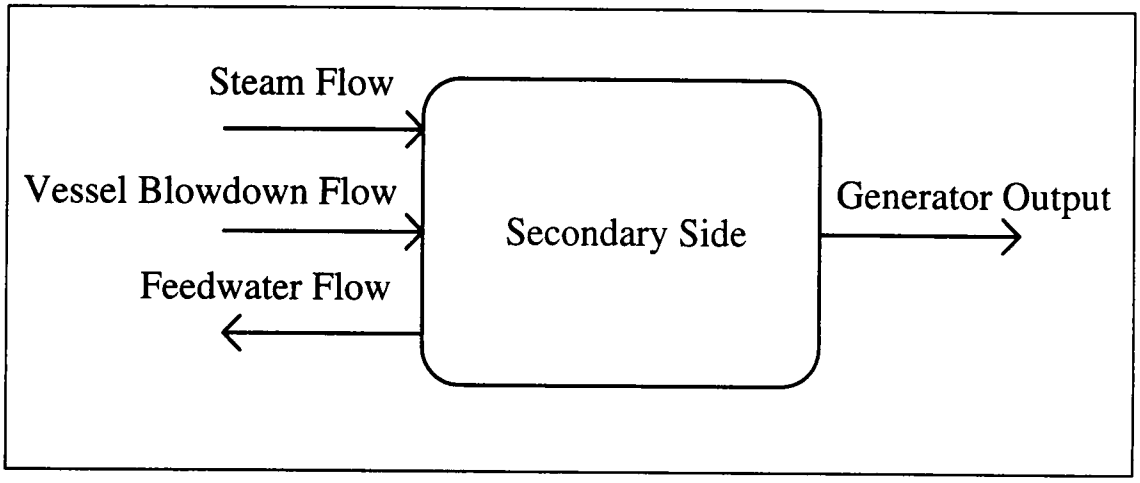


Figure 2.2: Global Schematic of the Secondary Side.

$$\beta = \frac{w_{vbd} h_{vbd}}{P} \qquad (2.3)$$

where

w_{vbd} : vessel blowdown mass flow rate

h_{vbd} : vessel blowdown enthalpy

Finally, the feedwater flow's contribution is given by

$$\gamma = \frac{w_{fw} h_{fw}}{P} \qquad (2.4)$$

where

w_{fw} : feedwater mass flow rate

h_{fw} : enthalpy of feedwater

A **C** computer code (see Appendix B) was written that calculates the gross heat rate of a typical four loop PWR plant using the above technique with the steam property routines (see Appendix A).

However, the methodology used in these calculations which is the approach based on classical thermodynamics, may produce erroneous results especially when the deviation from the ideal processes is high. Although, for most components, there are empirical curves which can be used to evaluate the deviation due to non-idealistic effects, these curves may also be affected by the degradation or aging of the components.

Neural Networks were applied to model the plant heat rate using actual data from an operating PWR. A total of fifteen signals from the plant were used as the input signals to the network. The patterns for which the unaccounted heat losses were very small were used as training patterns.

2.2.4 Feedwater Flow Monitoring

Since the operating thermal power of a nuclear power plant is limited by the Nuclear Regulatory Commission (NRC) licensing requirements, it is very beneficial to evaluate the thermal power accurately [21]. In the plant terminology, plant thermal power (Q_{th}) is the power calculated from the secondary side (see Figure 2.3) and is given by the following.

$$Q_{th} = Q_s + Q_{vbd} - Q_{fw} \tag{2.5}$$

$$Q_{th} = (w_{fw} - w_{vbd})h_s + w_{vbd}h_{vbd} - w_{fw}h_{fw} \tag{2.6}$$

$$Q_{th} = w_{fw}(h_s - h_{fw}) - w_{vbd}(h_s - h_{vbd}) \tag{2.7}$$

where

Q_{th} : thermal power transferred to the secondary side

w_{fw} : feedwater mass flow rate

w_s : steam mass flow rate

w_{vbd} : vessel blowdown mass flow rate

h_{fw} : enthalpy of feedwater

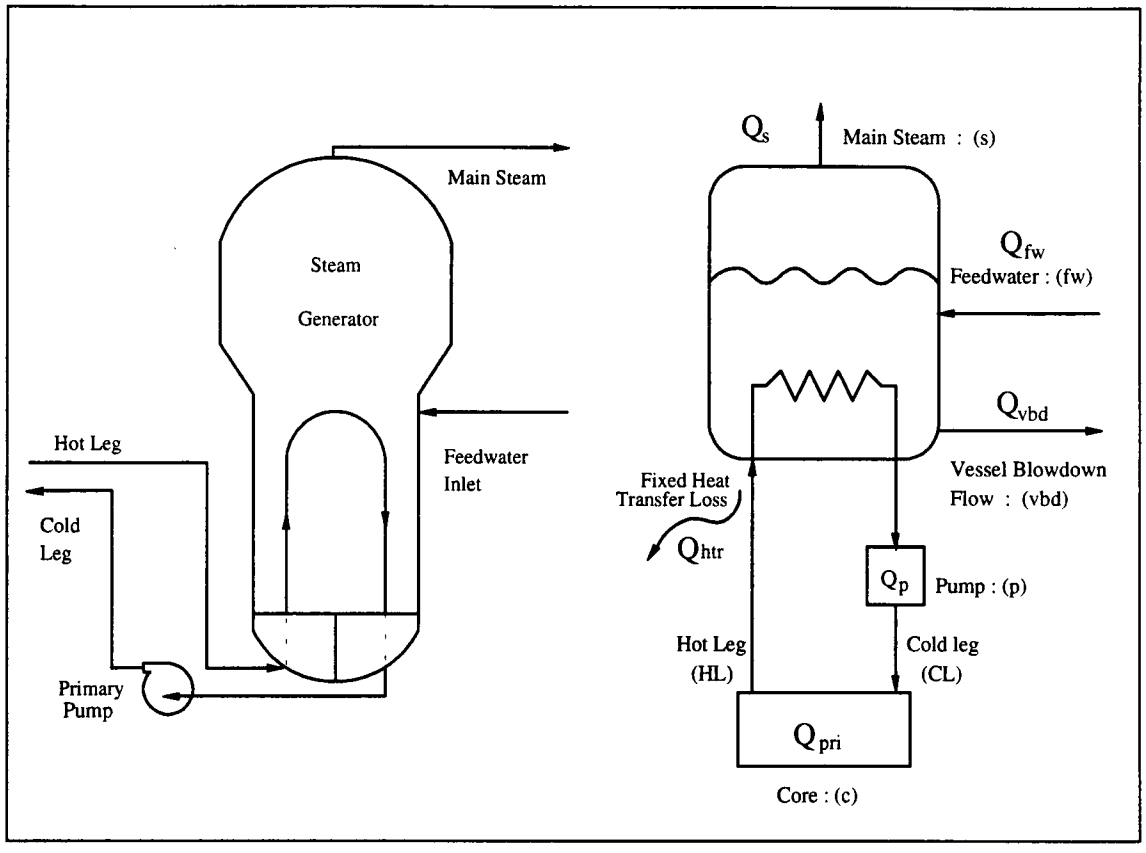


Figure 2.3: Schematic of a Steam Generator.

h_s : enthalpy of steam

h_{vbd} : enthalpy of vessel blowdown flow

The thermal power can be related to the primary side's thermal power (Q_{pri}) by the following.

$$Q_{th} = Q_{pri} + Q_p - Q_{htr} \qquad (2.8)$$

where

Q_p : pump work delivered to the primary coolant

Q_{htr} : heat lost by coolant by convection and radiation

Finally, the core thermal power is

$$Q_{pri} = w_{RCS}(h_{HL} - h_{CL}) \qquad (2.9)$$

where

Q_{pri} : core thermal power

w_{RCS} : primary coolant mass flow rate

h_{HL} : enthalpy of coolant in hot leg

h_{CL} : enthalpy of coolant in cold leg

The feedwater flow rate to the steam generators is one of the major signals used in determining the thermal power. This flow rate is measured by installed venturi meters (see Figure 2.4). There are alternative methods of measuring this flow rate, such as the use of ultrasonic flow meters.

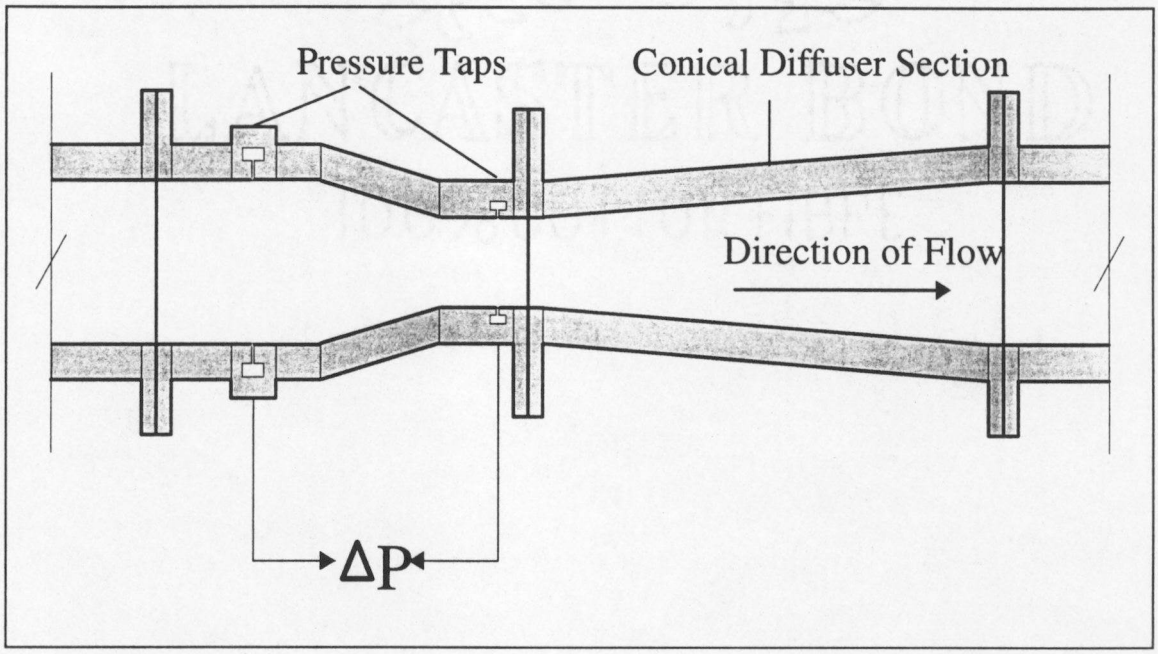


Figure 2.4: Schematic of a Venturi Meter.

There are two types of liquid ultrasonic flow meters [23]. In the first type, the so-called time of flight meters, a high frequency (1 MHz) pressure wave is beamed at an acute angle across the pipe. The time required for the wave to reach the opposite wall depends on whether it is moving with or against the flow and the speed of sound in the liquid. Flow rate information is obtained from the measurement of the travel time of the beam. In the second type, referred to as the Doppler flow meter, the pressure front does not traverse the pipe but is reflected back to a detector by particulate matter moving with the flow. The difference between the reflected frequency and the fixed transmitted frequency depends on the flow rate.

Time-of-flight flow meters are generally used in clean liquid applications such as in power plants, where the ultrasonic beam is not attenuated or continuously interrupted by fluid particles. There is a wide variety of ultrasonic flow meters. The notable differences are usually in the number of beam paths across the pipe area. A single path averages the velocity profile along the path not across the pipe area, that is why multi-path measurements are performed to reduce the velocity profile effects.

However, most power plants do not have ultrasonic flow meters as an on-line measurement system. The system is used as a precision test equipment whenever the feedwater flow values from the venturi meter become highly inconsistent with the heat balance calculations. The results are sometimes used to re-calibrate the venturi-meter.

2.2.4.1 The Venturi Meter

The venturi meter (see Figure 2.4) combines a short constricted portion between two tapered sections into a single unit and is usually inserted between two flanges in a pipe. Its purpose is to accelerate the fluid and temporarily lower its static pressure. Suitable pressure connections are provided for observing the difference in pressures between the inlet and the constricted portion, or throat.

Starting at the upstream flange, the first portion is a short cylindrical inlet which is a continuation of the upstream pipe line. This part is either machined inside or cast smooth so that its diameter can be determined accurately. The static pressure of the fluid at the inlet may be obtained through a single side-wall hole, or preferably there may be several holes distributed evenly around the inlet section.

Following the preliminary straight part is the entrance cone which has an included angle of about 20 degrees. The straight and the converging parts are connected by a curved surface. The entrance cone leads to the short cylindrical throat which is accurately machined and is provided with a side hole or holes for the measurement of the static pressure of the fluid at this section. The transition from the entrance cone into the straight throat is rounded off by a faired tangential curve to avoid the resistance caused by a sharp corner and also to preclude the possibility that the fluid might break away from the wall at high speeds and not fill the throat completely. The diameter of the throat is usually between one-third and three-fourths of the entrance or pipe line diameter.

The end of the throat leads, by another faired curve, into the exit or diffuser cone which has an included or total angle between 5 and 15 degrees. This terminates in the outlet flange for connecting the venturi to the pipe line.

Pressure drop across the venturi throat is measured and is related to the flow across the venturi. This relationship is given by the following formula [2].

$$w = 1890.07 d^2 F_a Y \frac{C_D}{\sqrt{1-\left(\frac{d}{D}\right)^2}} \sqrt{\rho \Delta P} \qquad (2.10)$$

where

w :	flow rate (lbm/hr)	Y :	Adiabatic expansion factor
d :	throat diameter (in)	C_D :	discharge coefficient
D :	pipe diameter (in)	ρ :	fluid density (lbm/ft^3)
F_a :	thermal expansion factor	ΔP :	pressure drop (psia)

Although the use of factors such as thermal expansion factor and adiabatic expansion factor in the conversion equation may introduce some errors, they can be eliminated by proper techniques such as sensor calibration. However, the most important problem in feedwater flow rate determination is the effect of deposition of small particles on the venturi converging section and throat over a period of time. Since this process changes the roughness of the throat surface, the calculated flow rates are higher than their actual values. This yields artificially high calculated plant thermal powers, thereby effectively derating the plant.

2.2.4.2 Venturi Fouling

Venturi fouling, or precipitation fouling in technical terms, may be defined as the phenomenon of a solid layer deposition on a surface arising primarily from the presence of dissolved inorganic salts in the flowing fluid which exhibit supersaturation under the process conditions [26]. Deposits formed under various conditions have different mechanical characteristics and are accordingly described by a rather loose terminology. The term "scaling" is generally used to describe a dense crystalline deposit well bonded to the metal surface. When the deposited layer is porous and loosely adherent, it is described by the term "soft scale" or "powdery deposit". Corrosion products such as iron oxides and iron hydroxides are common in feedwater flow.

An increase in the roughness of the converging or throat sections induces a lowering of the discharge coefficient (C_D) [27]. Since the discharge coefficient used in the conversion equations are higher than the actual coefficient, the measured feedwater flow rates become higher than the actual values. The effect of roughness on the convergent cone and the throat may be strikingly high.

However, accounting for the roughness change requires a knowledge of the height, spacing, configuration, and the geometrical shape of the roughness elements, which is almost impossible to evaluate [28].

2.2.4.3 Neural Network Models

An artificial neural network trained with reliable data can estimate the correct flow rate (used in thermal power calculations) regardless of the changes in the measurement system or the physical state of the feedwater [4]. The goal of this part of the study is to isolate the feedwater flow rate signal in order to estimate the correct flow rates, and to monitor possible fouling of venturi meters.

Feedwater flow estimation with neural networks was performed by using data from two different types of operating PWRs. Multi-input, single-output hetero-associative networks were built to predict the flow rate for each loop of the power plants. Input signals to the networks were determined by considering the heat and mass balances for the steam generator. The patterns immediately after refueling, when venturi tube is clean and is supposed to measure correct flow rates, were used as training patterns.

2.3 Thermal Performance Diagnostics

There has been a considerable effort in recent years to develop tools to assist plant operators in identifying and correcting thermal performance problems. Included among these tools are a series of performance diagnostic guides, in the form of both logic and decision trees, which can be used as tools for identifying the root causes of power plant thermal performance problems [18]. These logic and decision trees have been developed for both fossil and nuclear power plants. These trees represent a compilation of expert knowledge of the relationships between the behavior of observable process parameters

and the underlying problems that affect heat rate in power plants. As such, they provide valuable guidance in the techniques and steps to be used in troubleshooting performance problems at typical nuclear and fossil plants. However, in their present form, they can be difficult to interpret, to use, or to maintain.

An expert system approach is used to implement these trees. Expert systems can give the user a substantial insight into the problem through the rule base constructed from logic trees. The expert system can easily be modified for a new rule and it can be connected to the current data acquisition system to provide on-line diagnostics information.

On the other hand, application of a heat rate diagnostics expert system to a particular power plant requires the modifications specific to the plant since the trees mentioned above are generated for generic types of fossil or nuclear power plants. It is not easy or economical for utilities to have their own expert advisors.

In this thesis, an ANN methodology was proposed for identifying the component of the plant causing the degradation, and applied to a sample component which is one of the feedwater heaters of an operating PWR power plant. The idea is to design a neural network whose output is a measure of the thermal efficiency for each component of a power plant. Then, these networks can be used in diagnosing degradation of heat rates caused by one or more components.

The heat given by the extraction steam and the heat received by the feedwater can be used to determine the thermal performance of the system. A schematic of the fourth feedwater heater is given in Figure 2.5.

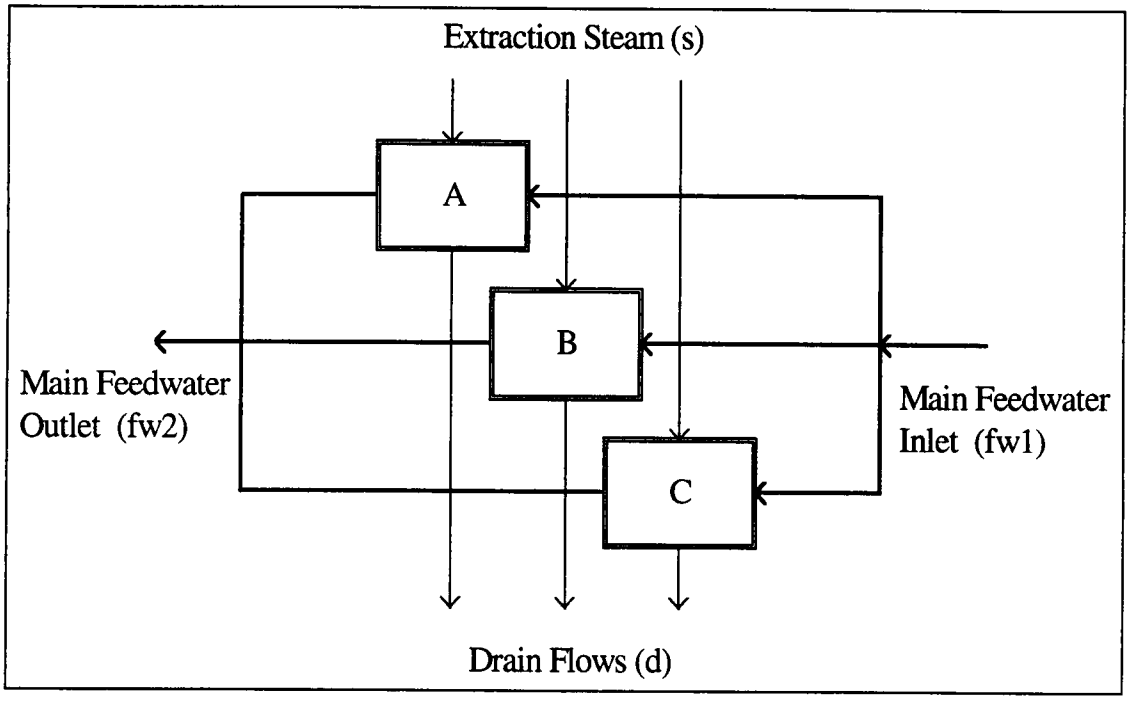


Figure 2.5: Schematic of a Typical Feedwater Heater.

Referring to Figure 2.5, an efficiency measure can be defined mathematically as

$$e = \frac{Q_{fw}}{Q_s} \tag{2.11}$$

where e is the efficiency of the component, Q_{fw} is the heat that is transferred to the main feedwater, and Q_S is the heat that extraction steam loses. Using the enthalpy equivalents of heats, the following is obtained.

$$e = \frac{w_{fw}(h_{fw2} - h_{fw1})}{w_s^A(h_s^A - h_d^A) + w_s^B(h_s^B - h_d^B) + w_s^C(h_s^C - h_d^C)} \tag{2.12}$$

where

w_{fw} : main feedwater mass flow rate

h_{fw1} : enthalpy of main feedwater at the entrance of feedwater heater

h_{fw2} : enthalpy of main feedwater at the exit of feedwater heater

w_s^j : steam mass flow rate entering section j=A,B,C

h_s^j : enthalpy of steam entering section j=A,B,C

h_d^j : enthalpy of drain flow exiting section j=A,B,C

Combining Equation (2.12) with the steam properties, the thermal performance of the component can be determined and monitored. Another technique for determining the thermal performance is to use neural networks for estimating the desired quantity. The thermal performance of the fourth feedwater heater was studied using the measurements related to this sub-system.

2.4 General Approach Using Neural Networks

This present study utilizes a multi-layer neural network methodology and the backpropagation (BP) training algorithm for the monitoring and the diagnostics of process variables in a system [8]. Neural networks are intrinsically parallel and non-algorithmic methods. The ability of the BP algorithm to learn any arbitrary nonlinear mapping from inputs to outputs, and the fault-tolerant property of a multi-layer network have a variety of applications [7].

The neural network is said to be heteroassociative when the input and the output layers are not identical. The BP algorithm computes the weights between pairs of processing elements so that the difference between the actual output and the network output is minimized in a least-squares sense. The output layer, in general, consists of several elements, but if the goal is to predict one variable as a function of others, then the output will be a single element. This algorithm can also be used for identical input and output sets for plant-wide monitoring purposes. In this case, the network is said to be auto-associative.

Chapter 3

ARTIFICIAL NEURAL NETWORKS

3.1 Introduction

This chapter discusses Artificial Neural Networks (ANN), particularly the Backpropagation (BP) algorithm as they relate to performance monitoring and diagnostics tasks applied to the balance-of-plant system of a PWR.

The human brain is probably the most complex computing device known. Most of its activities, like thinking, remembering, decision making, pattern recognition, and problem solving, have long been attracting people to construct computational schemes or devices that can emulate its operation [29].

The fundamental unit of human brain is the neuron (see Figure 3.1). Each neuron is a simple processing unit which receives and processes electrical signals on paths called dendrites from many other neurons and produces a single output signal which is sent to many other neurons by a path called axon. The transfer of information across the synaptic junction is chemical in nature, but has electrical side effects.

The brain and the spinal cord consist of approximately 100 billion highly inter-connected neurons. The axon of a neuron splits up and connected to dendrites of other neurons through junctions referred to as synapses. The transmission across this junction is chemical in nature and the synaptic efficiency is what is modified during learning.

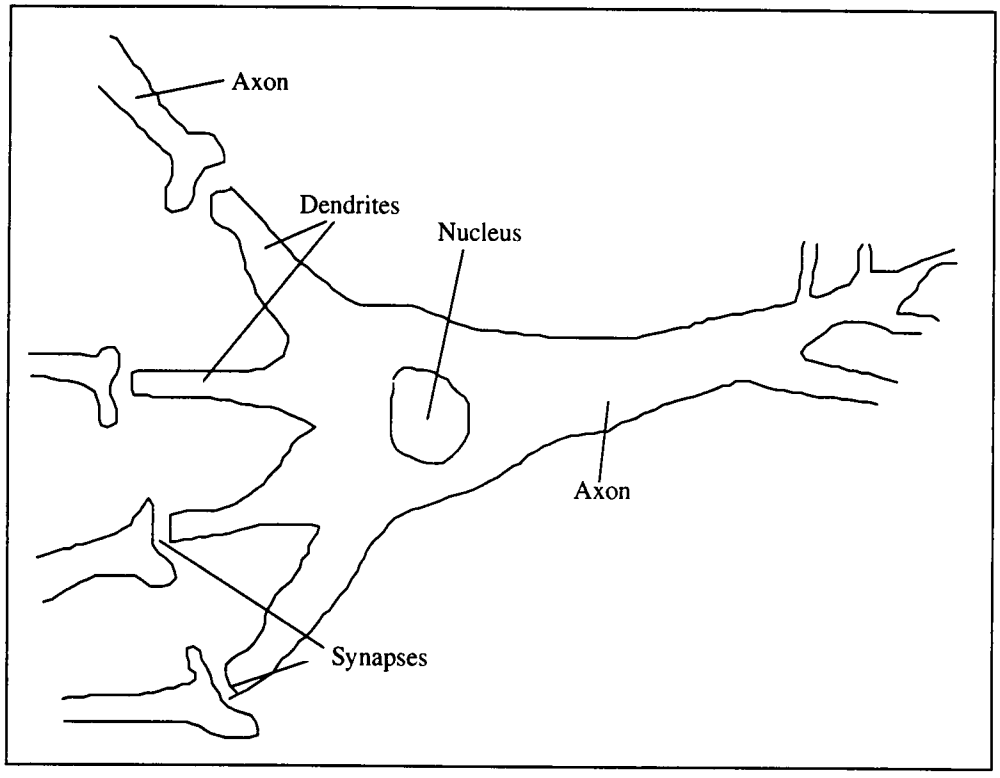


Figure 3.1: Schematic of a Neuron.

3.2 Artificial Neural Networks

Artificial neural networks are computational models inspired by the architecture of the nervous system of the human brain. In an artificial neural network, the unit analogous to the neuron is referred to as a "processing element" (PE) (see Figure 3.2). Each of the paths carrying input to processing elements has an adjustable weight value which is to be multiplied by the input values on the path. A PE has many input paths, also called the connections, and combines the weighted inputs usually by a simple summation. The combined input is then passed through a transfer or a squashing function which may be a tresholding function with two output states, or a continuous s-shaped function such as a sigmoid. The output value of the transfer function is usually passed directly to the output connection of the PE.

The output of a PE is connected to input paths of other PEs through the connection weights which are similar to the strength of the synapses. The output of each PE is multiplied by the corresponding connection weights before reaching another PE. Therefore, PEs combine the modified activation values from other elements.

An artificial neural network consists of many processing elements interconnected in the above manner. In general, PEs are grouped into layers or slabs which are designed in sequence from input to output layers. Then the connections are built between the neighboring layers. The input layer (sometimes called the input buffer) and the output layer, receive and transmit information to the outside, respectively. All other layers, so-called "hidden layers" , receive and transmit information within the network. A neural

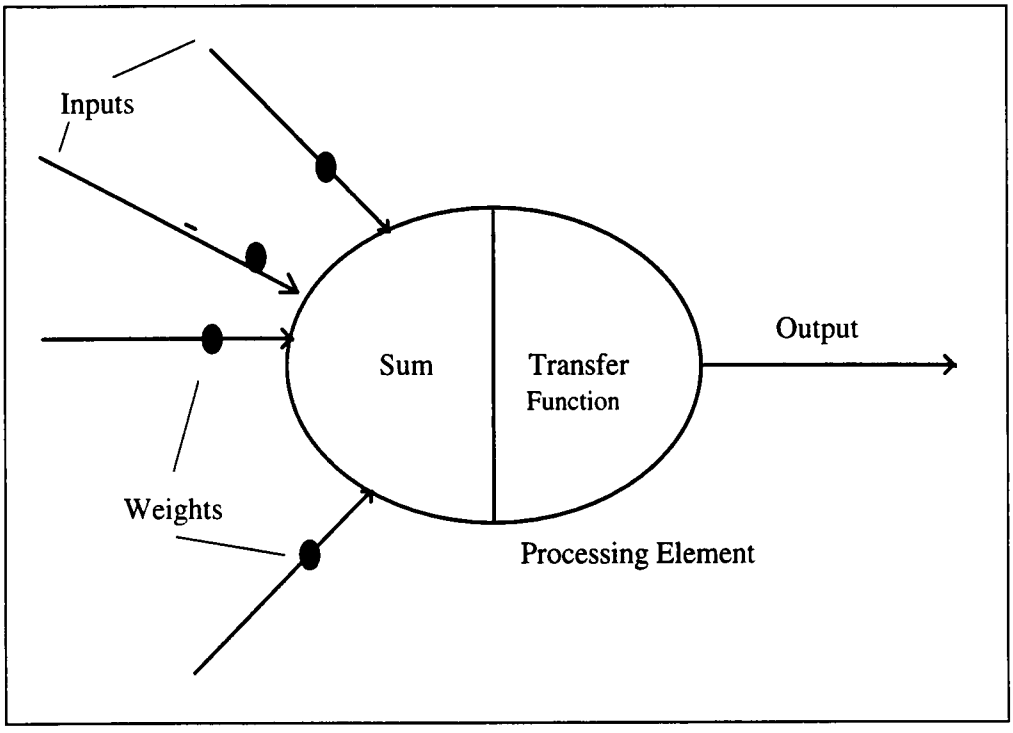


Figure 3.2: Schematic of a Typical Processing Element.

network is said to be auto-associative if the output and the input signals are identical. Otherwise, it is called a hetero-associative network.

Learning is one of the two main phases of operation in developing a neural network model. Learning is the process of adapting or modifying the connection weights between PEs as a result of the mismatch between the actual and the desired output of the neural network in response to an input presented to the input layer. In the case of supervised learning, the desired output for an input pattern is provided by the user. If there is no desired output , the learning is called unsupervised learning. Finally, if the teacher grades the network performance as being good or bad instead of giving a desired output, the learning process is called reinforcement learning.

One of the main characteristics of any network is its learning rule which specifies the way the connection weights are adjusted in response to an input pattern. Learning, in general, requires many presentations of the same pattern, and many presentations of all patterns which are to be taught. The parameters of a learning rule may change as the learning process continues.

The other main phase of network operation is the recall process. Recall is the process of propagating the input pattern presented to the input layer through the network to the output layer. In other words, recall is evaluating the response of a neural network to an input pattern.

If there are no feedback connections from one layer to another or to other PEs in the same layer, it is said to be a feedforward network. In this case, recall is performed in a

straightforward way from the input layer to the output layer, using weighted sums and the transfer functions of PEs.

3.3 Backpropagation Network Algorithm

3.3.1 General Properties

The multi-layer perceptron network is one of the most important historical developments in neural networks technology. It is a powerful mapping network that has been successfully applied to problems from financial analysis to signal processing [7] [8].

The multi-layer perceptron (MLP), is a multi-layer, feed-forward artificial neural network (see Figure 3.3). The capabilities of these networks come from the nonlinear transfer functions used on the processing elements. It is a well-known fact that a two-layer perceptron, with counting the input layer as a layer, forms half plane decision regions, and a three-layer perceptron can classify any convex region in the input space [29].

The backpropagation algorithm which will be discussed later utilizes a generalization of the least mean square algorithm. It uses a gradient descent technique to minimize the cost function which is the mean square difference between the desired and the actual network outputs. This gradient descent technique may find a local minimum in the cost function instead of the global minimum. Using extra hidden nodes, lowering the gain term used in adapting weights, adding momentum and restarting training with a different set of initial weights are some of the remedies used to avoid being trapped at a local minimum.

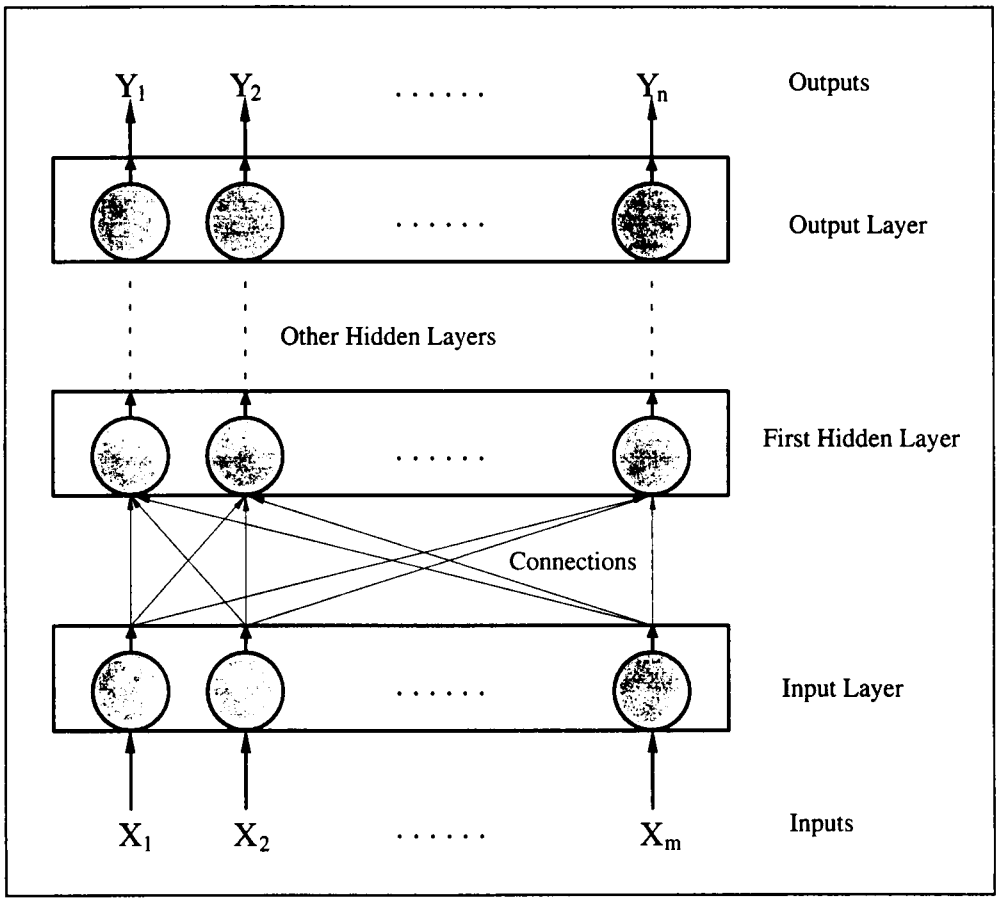


Figure 3.3: General Topology of a Multi-Layer Perceptron Network.

3.3.2 The Recall Process

The network architecture is a hierarchical design consisting of fully interconnected layers. A schematic showing this architecture is given in Figure 3.3. In general, it consists of L layers of processing elements numbered from the first layer to the last layer starting from 1. The first layer consists of **m** processing elements that simply receives the input pattern and fans it out without any processing to the next layer of elements. Each element in the second layer receives the output of all elements of the first layer, and processes the combined input according to the rules mentioned earlier, and passes the output to all of the elements of the next layer. This is repeated through all the layers of the network up to the last layer. The final layer consists of **n** processing elements and produces the response of the network to the input pattern.

3.3.3 The Training Process

The network's operation during the training phase consists of two sweeps through the network. The first step, or the forward step, starts by introducing a training pattern into the network's input layer. Using the procedure described in section 3.3.2, the network's response to this input pattern is evaluated. After having this estimate, the output elements of the network are compared with the correct (desired) output for the given input pattern. The second sweep (the backward pass) consists of computing the error for each PE in the output layer, and adjusting the incoming weights to each PE according to this error. The errors in each output PE are also used for assigning error to the processing elements of the layer below the output layer, and therefore, are used for adjusting the weights to the layer below the output. This process continues until the first layer is reached. The whole

process is repeated until the mean-square error between the desired and the actual network outputs decreases below a satisfactory level. The rigorous mathematical description of the training procedure is presented in the next section.

3.3.4 The Training Algorithm

A rigorous mathematical derivation of the BP algorithm can be found in many sources [7] [8] [29] [30]. Instead of the derivation, only the training procedure is given here. In order to avoid any confusion, the notation used in the expressions is given below.

x_j^p : is the output value of the jth node in layer p

w_{ji}^p : is the weight on connection from ith node in layer (p-1) to the jth node in layer p

I_j^p : is the weighted sum of inputs to the jth node in layer p

Then the weighted sum is evaluated by the following formula.

$$I_j^p = \sum_i w_{ji}^p x_i^{p-1} \tag{3.1}$$

The output of a node is given by the following.

$$x_j^p = f(I_j^p) \tag{3.2}$$

where the transfer function f may be a sigmoid (see Figure 3.4).

$$f(x) = \frac{1.0}{1.0 + e^{-x}} \tag{3.3}$$

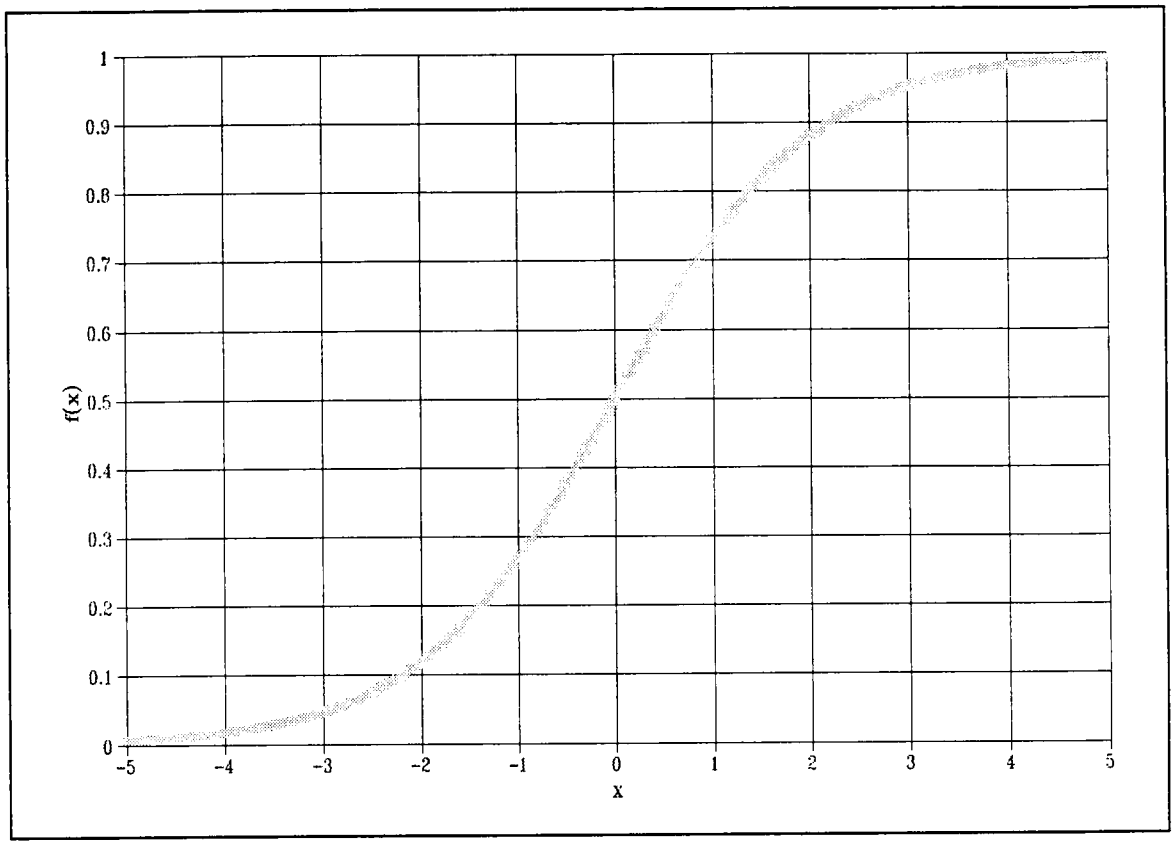


Figure 3.4: The Sigmoid Function. $f(x) = \frac{1}{1+e^{-x}}$

After presenting an input pattern, it is propagated through the network, and the output of the network is calculated based on the above equations. Assume that the output of the network is denoted by the vector **a**, and the desired output for the input pattern is denoted by the vector **d**. For each node of the output layer (o), the raw error (e_j^o) is calculated, and it is scaled by the derivative of the transfer function.

$$e_j^o = (d_j - a_j) f'(I_j^o) \tag{3.4}$$

Based on this scaled error on each node of the output layer, the connection weights from the layer (o-1) to the layer (o) are calculated according to the following equation.

$$w_{ji}^o(new) = w_{ji}^o(old) + \beta e_j^o x_j^{o-1} \tag{3.5}$$

where β is a constant and is called the learning coefficient.

For the layer (o-1) and all the other hidden layers, (p) in general, the scaled error for each of the nodes on the layer is evaluated based on the following formula.

$$e_j^p = f'(I_j^p) \sum_k (e_k^{p+1} w_{kj}^{p+1}) \tag{3.6}$$

Corresponding weight adjustments are made according to

$$w_{ji}^p(new) = w_{ji}^p(old) + \beta e_j^p x_j^{p-1} \tag{3.7}$$

So far, the discussion has been limited to the standard backpropagation algorithm. One of the problems of a gradient descent algorithm is setting an appropriate learning coefficient. Big learning coefficients may cause divergent behavior when the error surface is not

smooth. On the other hand, a small coefficient results in slow learning. The momentum term in the weight adjustment equation was introduced to solve this problem. Another method to accelerate the learning is to update the weights after some number of different input presentations instead of after each presentation. One other method is a variation of standard algorithm and is known as the fast backpropagation [35]. In this technique, the activation value in the weight adjustment equation is replaced by the activation value plus the error at the node.

3.4 Traditional versus Neural Network Models

The drawbacks of the classical techniques used in power plant thermal performance monitoring and diagnostics are discussed in the second chapter in detail. Basically, the thermodynamic calculations suffer from the need for empirical information on the performance of the components as a result of non-idealistic processes, and the expert systems need a complete and consistent rule base defining the plant for the sake of the inference process.

Unlike these methods, the neural network models are constructed by learning from examples in the form of input output patterns presented to the network. This is performed in a supervised learning mode. The network learns through the changes of connection weights as the actual output gets closer to the desired one.

The storage of information in the network is another distinguishing property of neural networks over other methods. The model information is distributed over all connection weights in the network. This makes the neural network models fault-tolerant to loss of

some part of the network. They are also tolerant to faulty, incomplete or noisy input information.

Finally, the modeling ability of the neural networks is not limited by the size or the complexity of the process. By setting appropriate network size, the number of hidden layers and the number of nodes, any complex, nonlinear mapping may be achieved by a neural network with an acceptable degree of error.

On the other hand, there are some disadvantages of network models. They cannot provide a physical insight into the problem unless they are designed to function so. Also, they cannot generate the reasons for producing a certain output. The determination of network parameters such as learning rates, momentum coefficients, and number of hidden nodes depends mostly on the specific application of interest and involves "trial and error". Erroneous results may be obtained if caution is not taken to avoid saturation and extrapolation beyond the training domain.

Chapter 4

NETWORK PARAMETERS AND DATA PREPARATION

1.1 Introduction

The purpose of this chapter is to discuss the determination of the parameters related to the neural network architecture, and the procedures used for implementing them. The success of using ANNs for any application depends highly on the data handling before or during network operation. Selection of input signals, data pre-processing, selection of training data are considered to be important steps in the process of applying a neural network.

Apart from the issues described above, there are some parameters to be determined which are directly related to the architecture of the network: The choice of the number of hidden layers, selecting the number of hidden layer nodes, setting learning and momentum coefficients.

4.2 Selection of Input Signals

For all the applications presented in this thesis, the governing physical phenomenon of the problem was taken into consideration to establish the process signals to be used as inputs to a given network for estimating a related parameter.

For auto-associative monitoring studies, almost all of the signals that are available and have importance were used as input, and also as output to the networks, since the purpose is to monitor the system as a whole rather than to monitor just one parameter.

Six signals were chosen for heat loss estimation. This was based on analytical calculations for heat losses from a typical weekly turbine cycle performance for PWRs [12].

In the case of feedwater flow rate analysis, a macroscopic heat and mass balance analysis on the steam generator was used to determine the signals that can be used to evaluate feedwater flow rate. A detailed discussion of this is presented in section 2.2.3.

On the other hand, the heat rate, being an overal indicator of the plant thermal performance, depends on almost every signal in the system. Fifteen signals from various parts of the system are used to estimate the heat rate. These signals include all major temperatures, pressures, and mass flow rates in the primary and the secondary side of the power plant.

The feasibility of applying neural networks for thermal performance diagnostics was studied for a feedwater heater system. The input signals determining the thermal efficiency of the system were determined from the analytical calculation (see section 2.3) of the component's thermal efficiency.

4.3 Data Preprocessing

Data preprocessing was found to be a very important issue in applying neural networks. Before creating a specific network for any of the tasks described so far, a visual analysis of the trends was made by observing their time plots. This was used to discard sample values that deviated from a normal range.

Because of the sigmoidal transfer functions (see Figure 3.4) used in processing elements, there is a large compression of the weighted sum if its magnitude is not close to zero. For the case of sigmoidal transfer function, the interval [-∞, +∞] is mapped into the interval [0, 1]. This function also goes to saturation very fast. Therefore, the engineering values for the signals which are in orders of tens, hundreds, or thousands will be mapped to very close values around the end points of the range.

Columnwise normalization was applied to overcome this effect in all studies. In the most general case, the minimum and the maximum of each signal were calculated and were used to linearly transform all values of that signal into a small interval such as [0.1, 0.9] as in the following formula.

$$x_{new} = -0.8\frac{x\max - x_{old}}{x\max - x\min} + 0.9 \qquad (4.1)$$

However, it was found that for some studies requiring extrapolation of data beyond the original input domain or output range, a smaller normalization interval, such as [0.2, 0.8] was used. This facilitates the ability of the network to extrapolate beyond the training domain.

4.4 Training Data

Selection of training data has a vital role in the performance of a supervised neural network. In general, for good recall performance, one should provide training patterns that fully span the range of input and output signals. However, it should always be emphasized that the training patterns which carry the input-output relationship for the system should be as accurate as possible.

In a nuclear power plant, the most precise data are expected to be recorded during the first weeks after refueling because there is a very intense maintenance program during outage for refueling. Almost all components, control systems, measurement systems are inspected, and corrective actions are taken if necessary.

Especially for feedwater flow estimation, it is very important to acquire the feedwater flow measurements early in the fuel cycle because of the fouling problem of venturi which is the on-line measurement system. Since venturi meters are inspected, cleaned and re-calibrated during plant outage, the data taken at the beginning of the fuel cycle must be very precise.

4.5 Steady-State versus Dynamic Data

Another subject in data processing that must be emphasized is the difference between the approach used in this thesis and in most of the previous neural network applications to power plant operations. Most of the previous applications used instantaneous data with some small sampling interval which can deal with fast and moderate-time dynamics of

the system. In the studies presented in this thesis, hourly or daily averaged signals were used and the long-term behavior of the system was analyzed.

The major drawback of using averaged values is the possible problems in the normalization process. Because of averaging, the effects due to fast transients are eliminated, and the input and output data ranges become very small compared to their mean values. Therefore, even one bad signal, a signal much below or much above the others can make normalization process very inefficient.

Therefore, a visual analysis of the input and the output signals was performed to make sure that the system does not have a pattern that is too high or too low with respect to the average of the corresponding signal. These types of patterns were removed during data preparation.

4.6 Number of Hidden Layers

Although it was believed that increasing the number of hidden layers is necessary for improving the training performance, the same goal can be achieved by keeping the number of layers at three (two hidden layers) and adjusting the number of processing elements of hidden layers. It has been shown that there exists a three-layer backpropagation neural network that can approximate any square-integrable function to within some ε mean-squared error accuracy [8].

In most of the networks implemented in this thesis, three-layer (one hidden layer) networks were used, and the determination of the number of nodes in the hidden layer was considered to be the main issue for training performance.

4.7 Number of Hidden Layer Nodes

The selection of the number of hidden layer nodes is an important issue. The number of hidden nodes determines the space separation abilities of a multi-layer perceptron. Insufficient number of hidden nodes may result in network's inability to solve the problem since the number of parameters in the model is less than necessary. In contrast, an excessive number of hidden nodes creates spurious decision boundaries or may result in overfitting of the model. It may also slow down the training process significantly.

There are some studies on the size of the hidden layer in a multi-layer network [31] [32]. For the early studies presented in the thesis, an empirical formula suggested by Eryurek [24] was used to get the number of hidden elements in a BP network. However, it was later found that the number generated by this formula is very high for steady-state or slowly varying processes. The procedure of starting with a low number of hidden units and increasing this number as learning problems occur was used.

4.8 Other Parameters

The transfer function used in all the applications is the standard sigmoid function given by

$$f(x) = \frac{1.0}{1.0 + e^{-x}}$$

Another critical parameter in the BP learning is the speed of convergence which is determined by the learning coefficient. In general, it is desirable to have fast learning, but not so fast as to cause instability of learning iterations. It is well known that starting with a large learning coefficient and reducing it as the learning process proceeds results in both fast learning and stability in iterations. This learning strategy was applied in the networks designed for the specific tasks.

The momentum coefficients were also set according to a schedule similar to the one for the learning coefficients. Cumulative update of the weights option was also used for increasing the learning performance. In this case, the delta weights, calculated by the training algorithm (see section 3.3.3), are accumulated for a specified number of input-output presentations instead of applying them after every presentation.

Chapter 5

NETWORK SENSITIVITY ANALYSIS

5.1 Introduction

A neural network can approximately represent most input-output mappings through a combination of connection weights between the nodes. It has been shown that a three-layer perceptron with a monotonically increasing transfer function such as sigmoid function can realize any continuous mapping [8]. The input-output model by a neural network may be used to determine the sensitivity of the output to any of the input variables. These sensitivities may then be used to establish the choice of signals for estimating the network output.

Differential and statistical sensitivity measures were applied to the neural networks developed for feedwater flow rate estimation. An analytical derivation (given in the next section) shows that differential sensitivity is a first order approximation to the statistical sensitivity. This chapter presents the definitions and related expressions for evaluating the network's output sensitivity by differential and statistical measures.

5.2 Differential Sensitivity Measure

In system theory, the model sensitivity is defined as the partial derivative of the output with respect to an input or a related parameter of interest. The output sensitivity of a single-output, three-layer neural network with a differentiable transfer function, with

respect to any of its inputs, is presented here (see Figure 5.1). The following list gives the terminology used in both differential and statistical sensitivity derivations.

in	input layer
h	hidden layer
o	output layer
f	transfer function
x_j^p	output of i-th node at p-th layer
w_{ji}^p	connection weight from i-th node at (p-1)-th layer to the j-th node at p-th layer
I_j^p	weighted sum to the j-th node on layer p

The differential sensitivity is defined as

$$S(y^o, x_i^{in}) = \frac{\partial y^o}{\partial x_i^{in}} \tag{5.1}$$

Applying the chain rule for derivatives gives the following.

$$\frac{\partial y^o}{\partial x_i^{in}} = \frac{\partial y^o}{\partial I^o} \frac{\partial I^o}{\partial x_i^{in}} \tag{5.2}$$

where

$$I^o = \sum_j w_j^o x_j^h \tag{5.3}$$

Substituting for the weighted sum, the above equation becomes

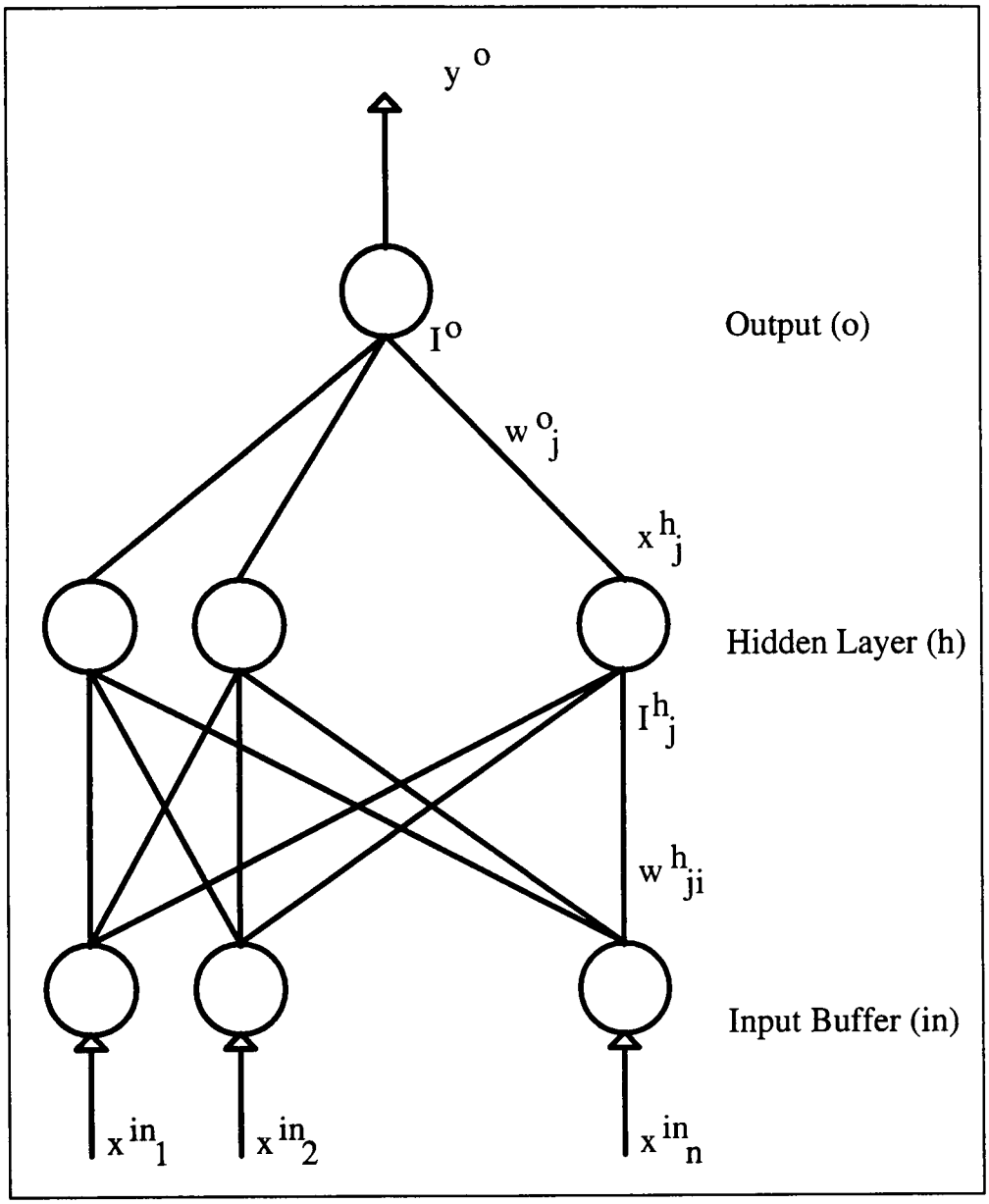


Figure 5.1: A Three-Layer Single-Output Neural Network.

$$\frac{\partial y^o}{\partial x_i^{in}} = \frac{\partial y^o}{\partial I^o} \sum_j w_j^o \frac{\partial x_j^h}{\partial x_i^{in}} \qquad (5.4)$$

Applying the chain rule again for x^h_j gives

$$\frac{\partial y^o}{\partial x_i^{in}} = \frac{\partial y^o}{\partial I^o} \sum_j w_j^o \frac{\partial x_j^h}{\partial I_j^h} \frac{\partial I_j^h}{\partial x_i^{in}} \qquad (5.5)$$

Using the following expressions

$$\frac{\partial I_j^h}{\partial x_i^{in}} = w_{ji}^h \quad , \quad \frac{\partial y^o}{\partial I^o} = f'(I^o) \quad ,and \quad \frac{\partial x_j^h}{\partial I_j^h} = f'(I^h{}_j)$$

the expression for differential sensitivity is obtained

$$S(y^o, x_i^{in}) = f'(I^o) \sum_j w_j^o f'(I_j^h) w_{ji}^h \qquad (5.6)$$

5.3 Statistical Sensitivity Measure

The statistical sensitivity of output to an input is the ratio between the standard deviation of the output and the standard deviation of the input [33] [34]. Before evaluating this quantity, the delta change in the output as a result of a delta change in the input is evaluated and is used for the sensitivity.

Let

$$x_i^{in} = x_{is}^{in} + \delta x_i^{in} \qquad (5.7)$$

where x_{is}^{in} is the steady-state value of x_i^{in}, and δx_i^{in} is some perturbation to the signal around the steady-state value. If this signal is presented to the network, the change in the weighted sums at the hidden layer processing elements can be expressed as

$$I_j^h = \sum_j w_{ji}^h x_i^{in} \tag{5.8}$$

$$\delta I_j^h = w_{ji}^h \delta x_i^{in} \tag{5.9}$$

The change in the output of the hidden layer nodes is expressed as

$$x_j^h = f(I_j^h)$$

$$\delta x_j^h = \sum_{k=1}^{\infty} \frac{\partial f^{(k)}(I_j^h)}{\partial I_j^h} \frac{(\delta I_j^h)^k}{k!} \tag{5.10}$$

A first order approximation is

$$\delta x_j^h = \frac{\partial f(I_j^h)}{\partial I_j^h} \delta I_j^h \tag{5.11}$$

The above relationship is used to evaluate the change in the weighted sum of the output node.

$$I^o = \sum_j w_j^o x_j^h$$

$$\delta I^o = \sum_j w_j^o \delta x_j^h \tag{5.12}$$

With the above preparation, the change in the output as a result of input perturbation is evaluated

$$y^o = f(I^o)$$

$$\delta y^o \cong \frac{\partial f(I^o)}{\partial I^o}\delta I^o$$

$$\delta y^o = \left\{ f'(I^o)\sum_j w_j^o f'(I_i^h) w_{ji}^h \right\}\delta x_i^{in} \qquad (5.13)$$

Now, consider δx^{in}_i as a random variable with zero mean. The statistical sensitivity is defined as [33]

$$S(y^o, x_i^{in}) = \lim_{\sigma_x \to 0}\frac{\sigma_y}{\sigma_x} = \lim_{\sigma_x \to 0}\frac{\left\{E\left[(y^o - y_s^o)^2\right]\right\}^{1/2}}{\left\{E\left[(x_i^{in} - x_{is}^{in})^2\right]\right\}^{1/2}} \qquad (5.14)$$

where y^o_s and x^{in}_{is} are the averages of the output and the input signals respectively.

$$S(y^o, x_i^{in}) = \lim_{\delta x_i^{in} \to 0}\left\{\frac{E\left[(\delta y^o)^2\right]}{E\left[(\delta x_i^{in})^2\right]}\right\}^{1/2} \qquad (5.15)$$

Using Equation (5.13) for δy in terms of δx gives

$$S(y^o, x_i^{in}) = f'(I^o)\sum_j w_j^o f'(I_j^h) w_{ji}^h \qquad (5.16)$$

Comparing this result with the expression for differential output sensitivity shows that the differential sensitivity is a first order approximation to the statistical sensitivity. In this thesis both measures were used, and the results were compared.

For the sensitivity studies in this thesis, experimental evaluation of sensitivities was performed for both differential and statistical measures, and they were found to be very close. For the differential sensitivity, different perturbations were applied to the input, and the change of the output was evaluated through network recall. The ratio of the two as the input perturbation goes to zero gives the differential sensitivity. For the statistical sensitivity, different Gaussian sequences with zero mean and known standard deviations were applied to the input, and the standard deviation of the output was calculated using the results of network recall. The ratio of the two as the input standard deviation goes to zero gives the statistical sensitivity of the output.

Chapter 6

RESULTS AND DISCUSSION

6.1 Introduction

The task of power plant performance monitoring and diagnostics requires the estimation of certain variables or parameters of the system and developing techniques for detecting the degradations in the system's performance. Currently, analytical and empirical methods are used for this purpose. Some power plants also use rule-based modeling combined with classical techniques for monitoring, and especially for diagnostics purposes.

The multi-layer perceptron which uses the backpropagation learning algorithm was applied in some of the important areas of nuclear power plant performance monitoring and diagnostics, with the focus on the balance-of-plant system. The applications presented in this thesis are

1. Process pattern monitoring using autoassociative networks
2. Heat loss estimation
3. Heat rate estimation
4. Feedwater flow rate estimation.

The analyses were performed using actual data from two different types of operating PWRs. A commercial software package called **NWORKS** was used for building neural

networks in all the studies presented in this thesis [35]. This chapter presents the results obtained from these studies.

Each section starts with an introduction to the specific area of application and then presents the topological information about the networks. Data preparation is described briefly and the results are given in the form of plots. Finally, a discussion is given on the results obtained from the study.

6.2 Process Pattern Monitoring Using Auto-associative Networks

Auto-associative monitoring with neural networks has been investigated and found to be successful in many applications. In this application, input and output layers are identical, each with several processing elements. Therefore, the network is trained to repeat its input at the output which provides the ability to detect any changes in the state of the system or the degradations in one or more plant signals. This is because of the fact that each state of a process can be represented by the patterns which are in a small region of the hyper-space spanned by the input signals. If any of the signals change during operation, the input pattern will change accordingly forming a new pattern.

The data used for this analysis were taken from weekly turbine cycle performance report data files of an operating four loop Westinghouse PWR. A maximum of two patterns per week were available. These are hourly averages of the related signals. All the signals in the data were normalized into the interval [0.2, 0.8]. There were a total of 41 patterns, and 15 of them were used for training the network. Most of the signals were from the BOP side of the PWR.

A three-layer (one hidden layer) network was built to monitor various signals in this PWR. Sixteen signals are selected as inputs to the network. These are listed in Table 6.1.

Therefore, the network has 16 input and 16 output nodes. The number of hidden layer elements is 48.

The network was trained up to a learning root-mean-square error of 10^{-3} in the normalized domain, and then used for recalling all the patterns. The results comparing the measured and the estimated values of some of the signals are given in Figures 6.1 through 6.8. Learning and recall root-mean-square errors (RMSE) are given in Table 6.2. RMSE is used as a performance criterion in the applications and is given by the following formula.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{mea}(i) - x_{est}(i)\right)^2} \qquad (6.1)$$

where

N : total number of patterns

x_{mea} : measured value of the variable

x_{est} : estimated value of the variable.

After analyzing the results of this study, it was found that none of the signals had any form of anomaly, thus no abnormal operating condition was observed during that period according to the neural network model.

Table 6.1: List of Signals for Auto-associative Monitoring.

#	SIGNAL	UNIT
1	Feedwater Temperature	^{o}F
2	Main Steam to MSR A1	Psig
3	Impulse Pressure PT-1-73	Psia
4	Impulse Pressure PT-1-72	Psia
5	Average CCW Inlet Temperature	^{o}F
6	Condenser Backpressure	inches Hg
7	Steam Throttle Pressure	Psia
8	RCS Pressure	Psia
9	Cold Leg Temperature	^{o}F
10	Hot Leg Temperature	^{o}F
11	Vessel Blowdown Flow	GPM
12	Feedwater Pressure	Psia
13	Feedwater Flow Rate	lbm/hr
14	Corrected Heat Rate	Btu/kWh
15	MSR Steam to LPT A Temperature	^{o}F
16	Steam Generator Pressure	Psia

where

MSR : moisture separator reheater

CCW: condenser cooling water

RCS : reactor coolant system

LPT : low pressure turbine

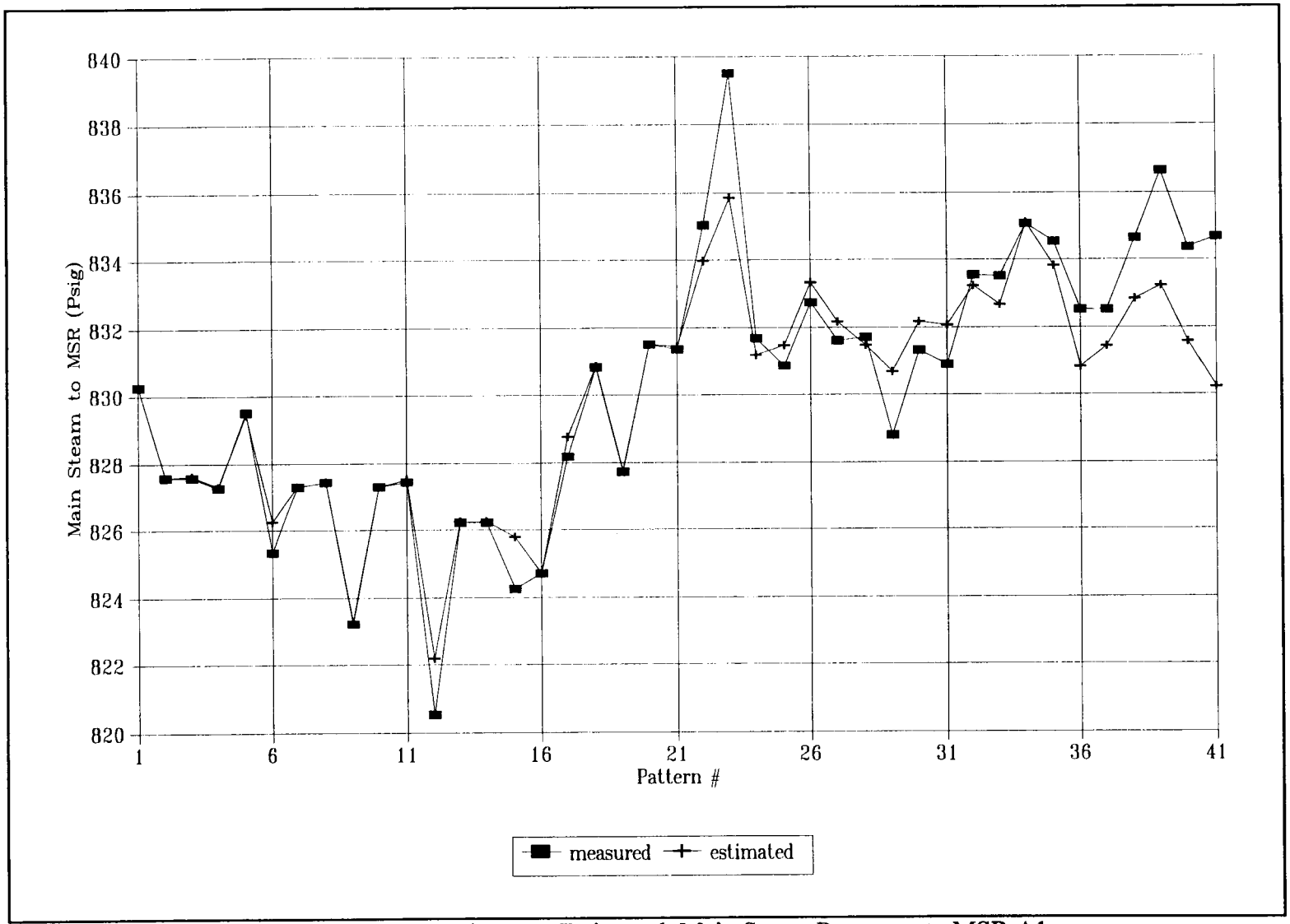


Figure 6.1: Measured versus Estimated Main Steam Pressures to MSR A1.

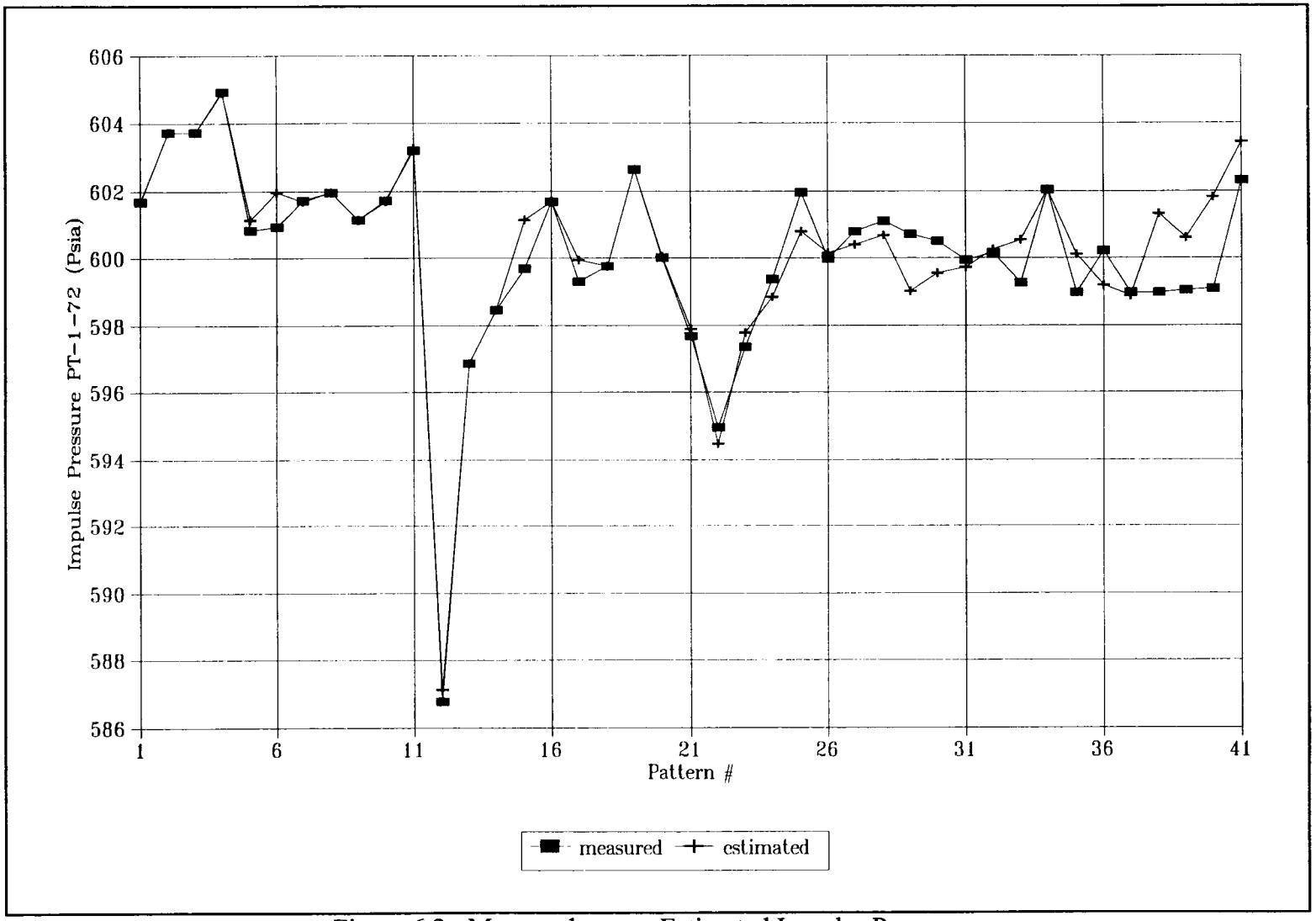


Figure 6.2: Measured versus Estimated Impulse Pressures.

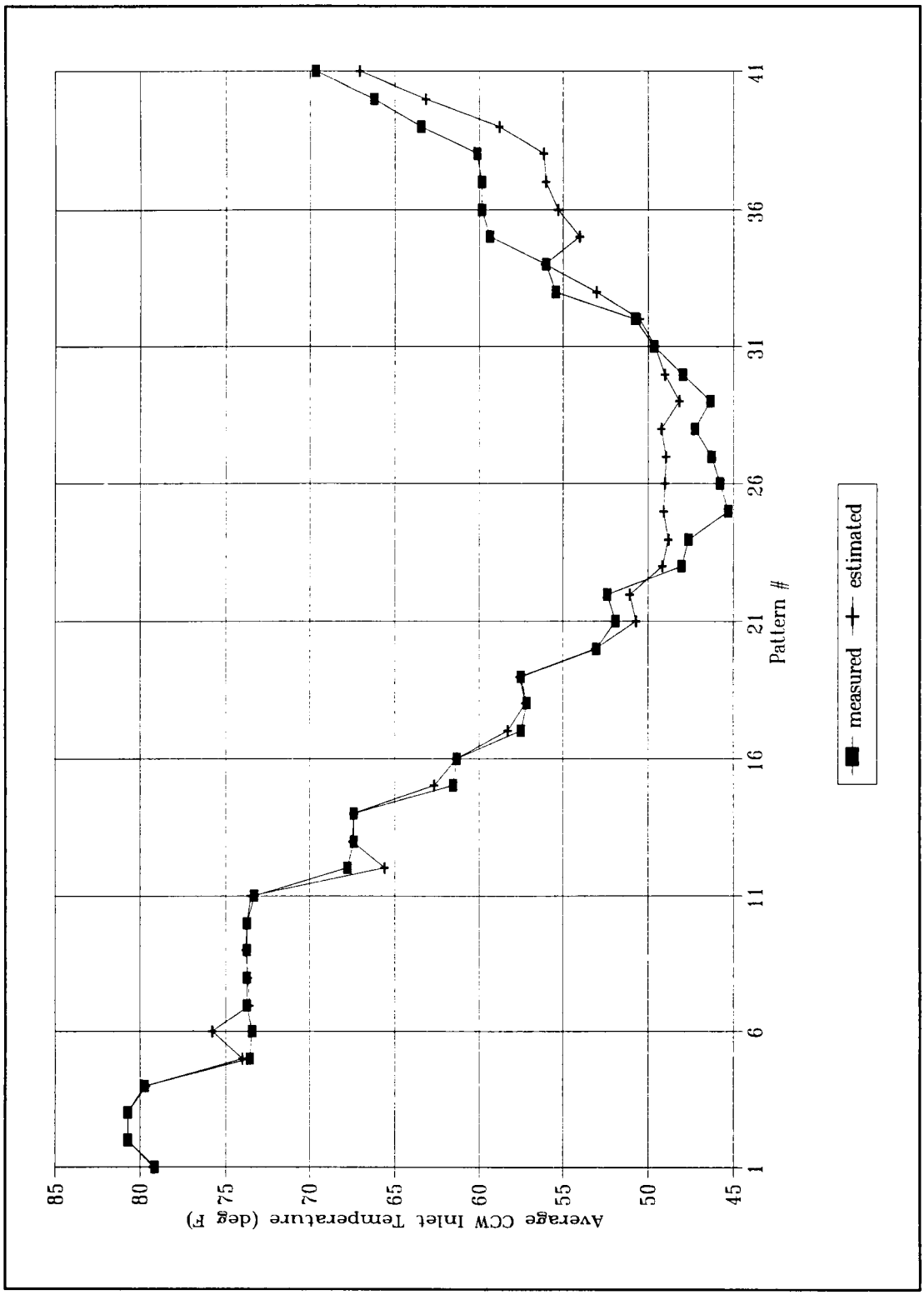


Figure 6.3: Measured versus Estimated CCW Inlet Temperatures.

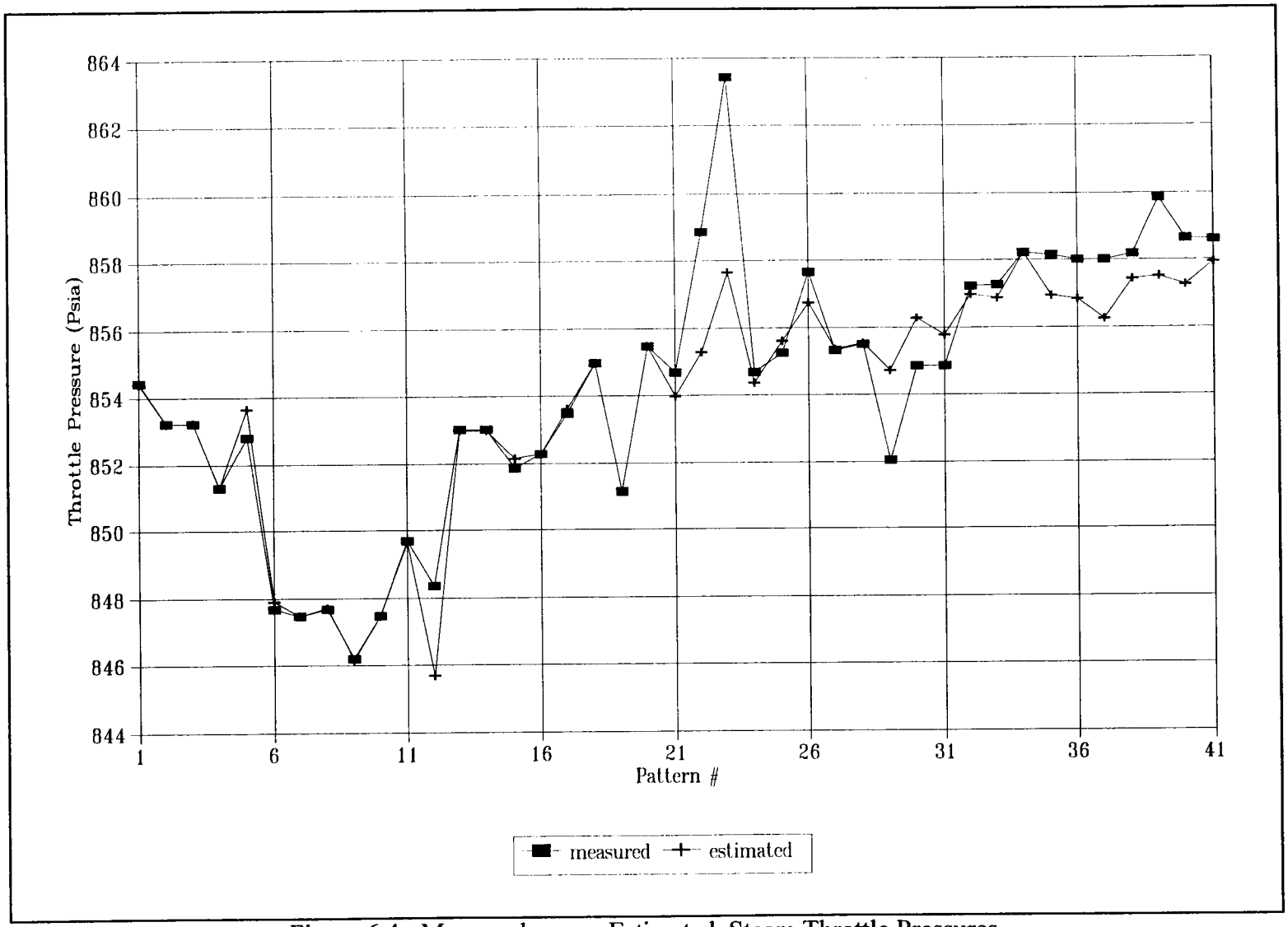


Figure 6.4: Measured versus Estimated Steam Throttle Pressures.

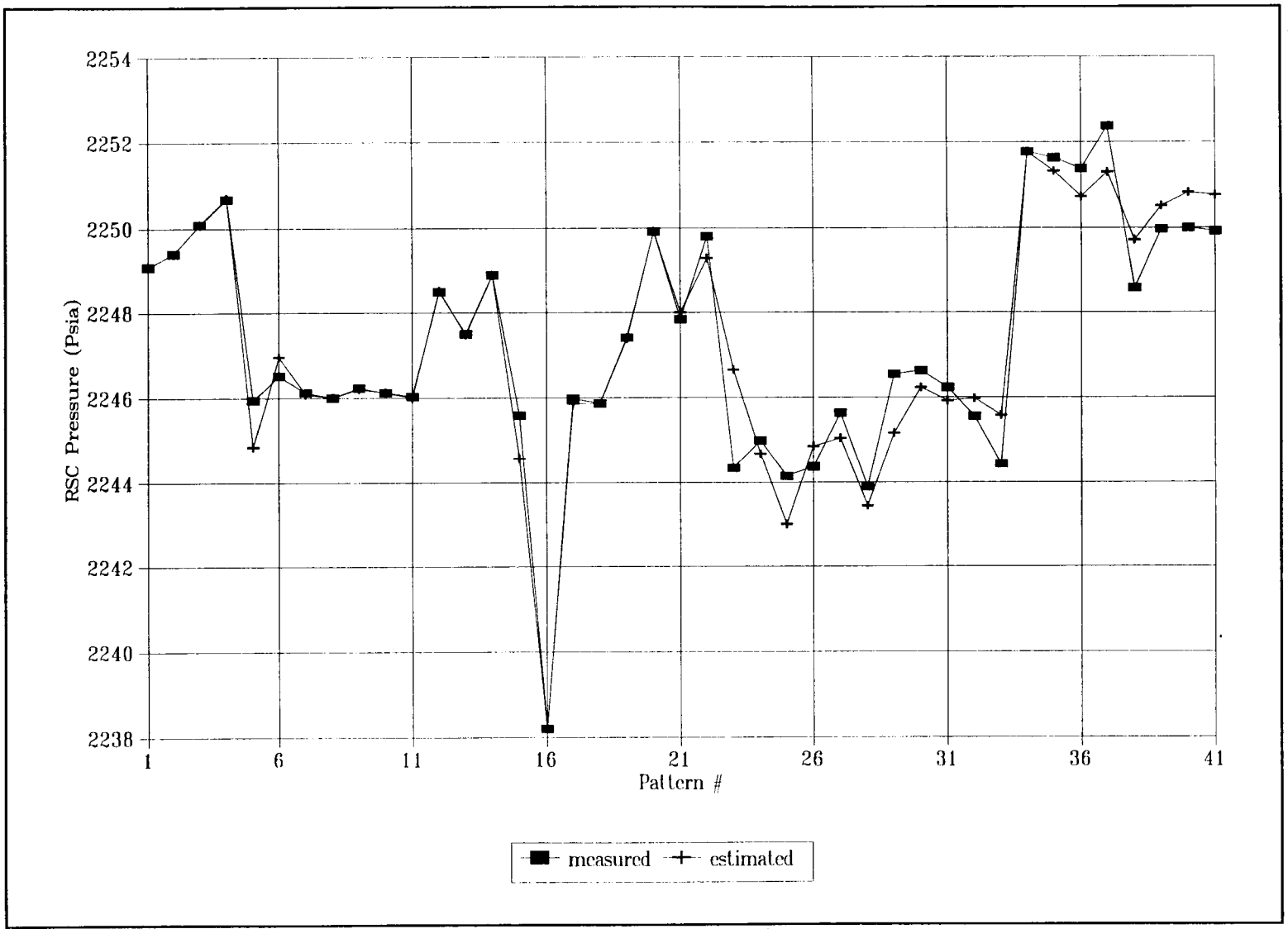


Figure 6.5: Measured versus Estimated RCS Pressures.

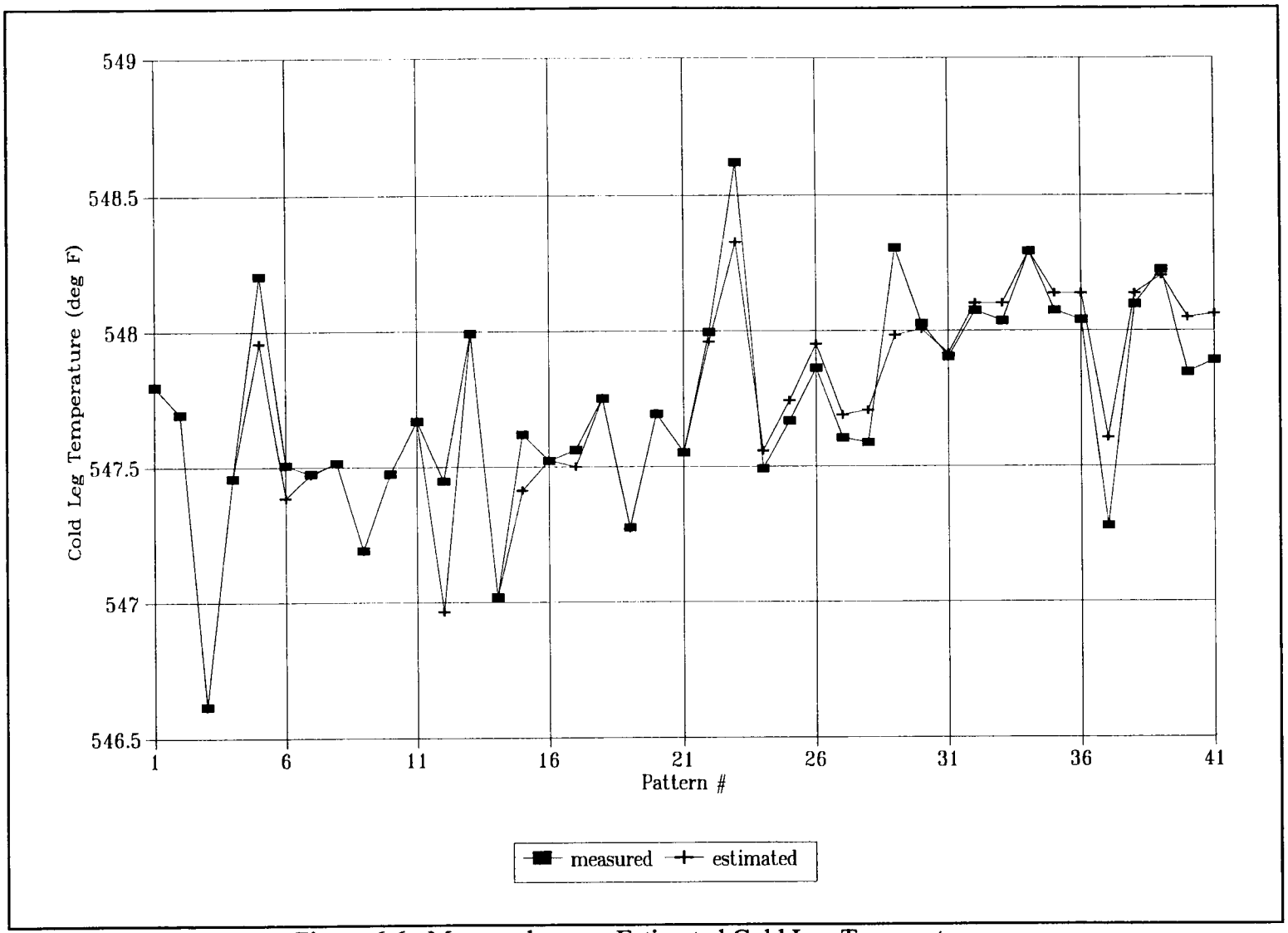


Figure 6.6: Measured versus Estimated Cold Leg Temperatures.

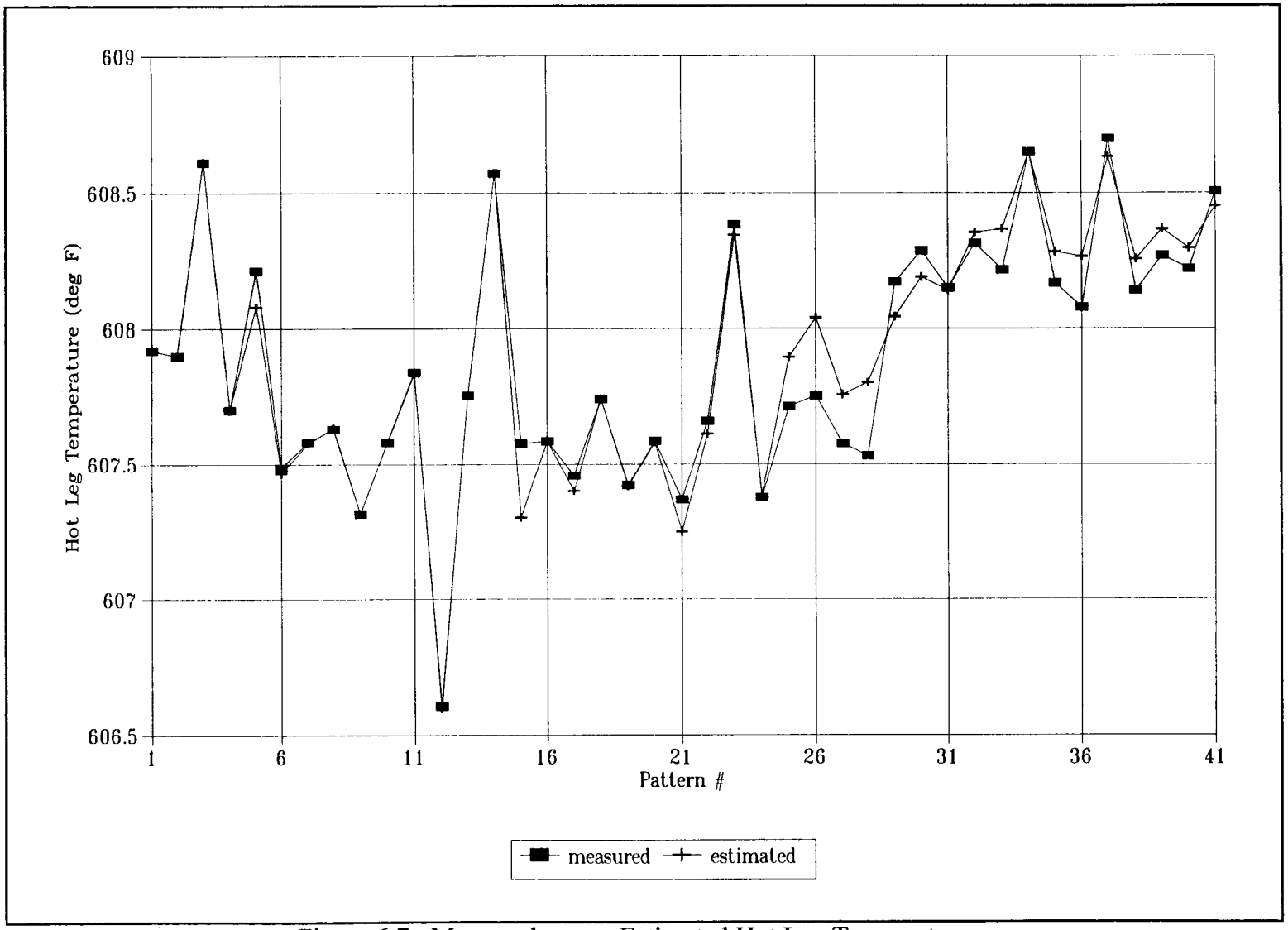


Figure 6.7: Measured versus Estimated Hot Leg Temperatures.

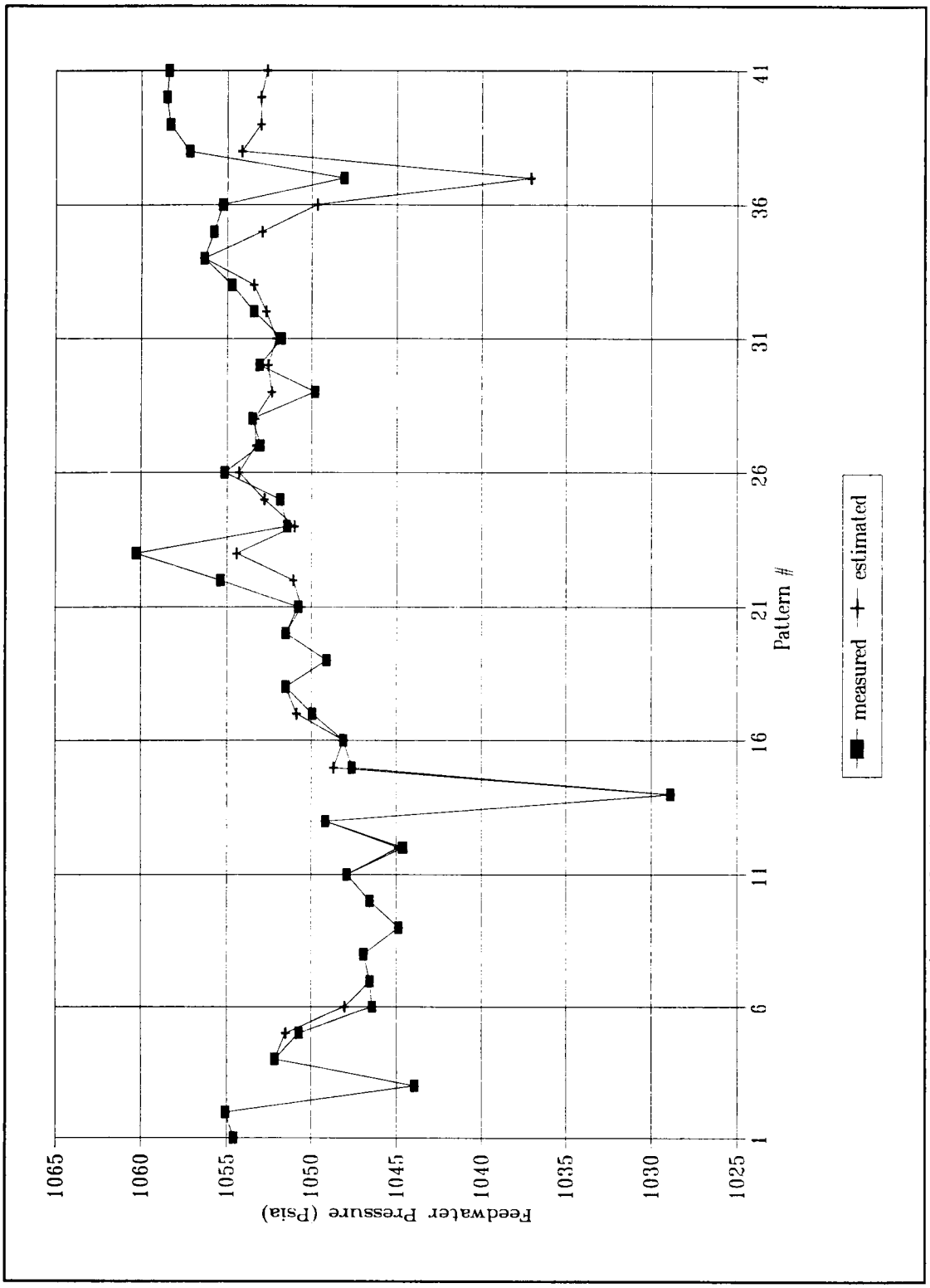


Figure 6.8: Measured versus Estimated Feedwater Pressures.

Table 6.2: Learning and Recall Root-Mean-Square Errors (RMSE).

#	SIGNAL	LEARNING RMSE	RECALL RMSE	UNIT
1	Feedwater Temperature	0.045	0.83	^oF
2	Main Steam to MSR A1	0.184	1.16	Psig
3	Impulse Pressure PT-1-73	0.179	0.83	Psia
4	Impulse Pressure PT-1-72	0.138	0.93	Psia
5	Average CCW Inlet Temperature	0.255	1.45	^oF
6	Condenser Backpressure	0.032	0.03	inches Hg
7	Steam Throttle Pressure	0.134	1.18	Psia
8	RCS Pressure	0.105	0.81	Psia
9	Cold Leg Temperature	0.055	0.37	^oF
10	Hot Leg Temperature	0.045	0.30	^oF
11	Vessel Blowdown Flow	0.293	4.63	GPM
12	Feedwater Pressure	0.084	1.68	Psia
13	Feedwater Flow Rate	5.619	54.55	lbm/hr
14	Corrected Heat Rate	0.164	4.28	Btu/kWh
15	MSR Steam to LPT A Temp.	0.032	0.43	^oF
16	Steam Generator Pressure	0.114	1.16	Psia

MSR : moisture separator reheater

CCW: condenser cooling water

RCS : reactor coolant system

LPT : low pressure turbine

6.3 Heat Loss Estimation

One of the main objectives of power plant operation is to minimize the heat losses in the system. This quantity is defined as the difference between the measured generator output and the expected plant electrical output. Some portion of these losses are due to factors, such as use of steam for other purposes, and environmental effects such as changes in river temperature. These can be evaluated analytically. On the other hand, the so-called unaccounted heat losses are due to factors such as high pressure turbine efficiency and other parameters which are extremely difficult to calculate accurately.

The primary objective of this work is to estimate the total unaccounted heat losses. Because of the differences in the unaccounted loss trends of the two units, this analysis was performed for each unit separately. In the original data files, there were a total of 78 patterns for the first unit and 58 patterns for the second unit. For each unit, the data points where the unaccounted heat losses are less than 1.0% of the plant output power are extracted to be the training patterns for neural networks. Since a power loss less than this value is reasonable, the patterns on these days are supposed to carry the correct relationship among the signals. A total of 24 points for the first unit and 25 points for the second unit were selected as training patterns for the analysis.

For the prediction of this loss, hetero-associative neural networks were used with the output layer having one processing element representing the unaccounted heat loss. Six signals from the plant (see Table 6.3) were selected as the input signals to the networks. These are the most important signals used in the plant calculations for estimating heat losses [12].

Table 6.3: List of Signals Used for Heat Loss Estimation.

#	SIGNAL	UNIT
1	Reactor Average Loop ΔT	^{o}F
2	Feedwater Mass Flow Rate	lbm/h
3	Feedwater Temperature	^{o}F
4	Average Impulse Pressure	Psia
5	CCW Inlet Temperature	^{o}F
6	Condenser Backpressure	inches Hg

While performing an estimation analysis, a parametric study on the number of hidden layer nodes was also made. Four different three-layer BP networks with 5, 10, 15, and 20 hidden layer processing elements were developed for each unit of the plant. These networks were trained using 5000 iterations for a pattern. Figures 6.9 and 6.10 present the results of training and recall phases for the first unit. Figures 6.11 and 6.12 present the results of training and recall phases for the second unit. Learning and recall root-mean-square errors are given in Tables 6.4 and 6.5.

The estimated unaccounted heat losses show big differences from the values calculated for the weekly turbine cycle reports for the first unit. The estimated values of the total unaccounted loss for the second unit is reasonable, whereas they are very high for the first unit. Actually, the two units have very different trends for this loss. The unaccounted heat loss increases steadily in the first unit, and it has an oscillatory pattern in the second unit. The results of neural networks have shown that the results of calculational method is not consistent throughout the data period.

6.4 Heat Rate Estimation

Plant heat rate depends on many process variables in the system. This quantity is defined as the amount of heat supplied to the secondary side of the plant to produce one kWh electricity. Current heat rate calculations are based on the thermodynamic methods using some empirical data and steam tables. This section presents the results of plant heat rate estimation using neural networks.

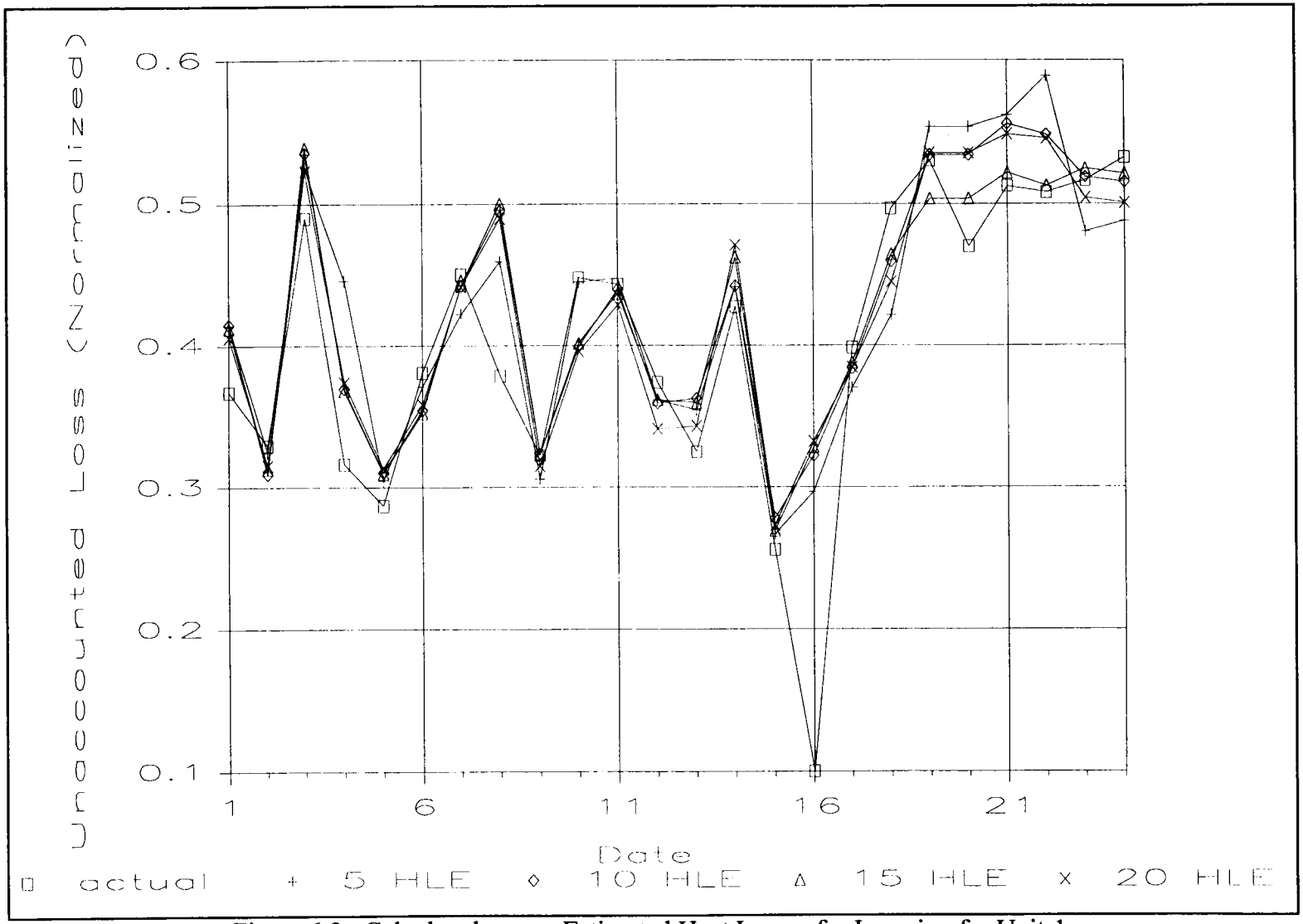


Figure 6.9: Calculated versus Estimated Heat Losses for Learning for Unit-1.

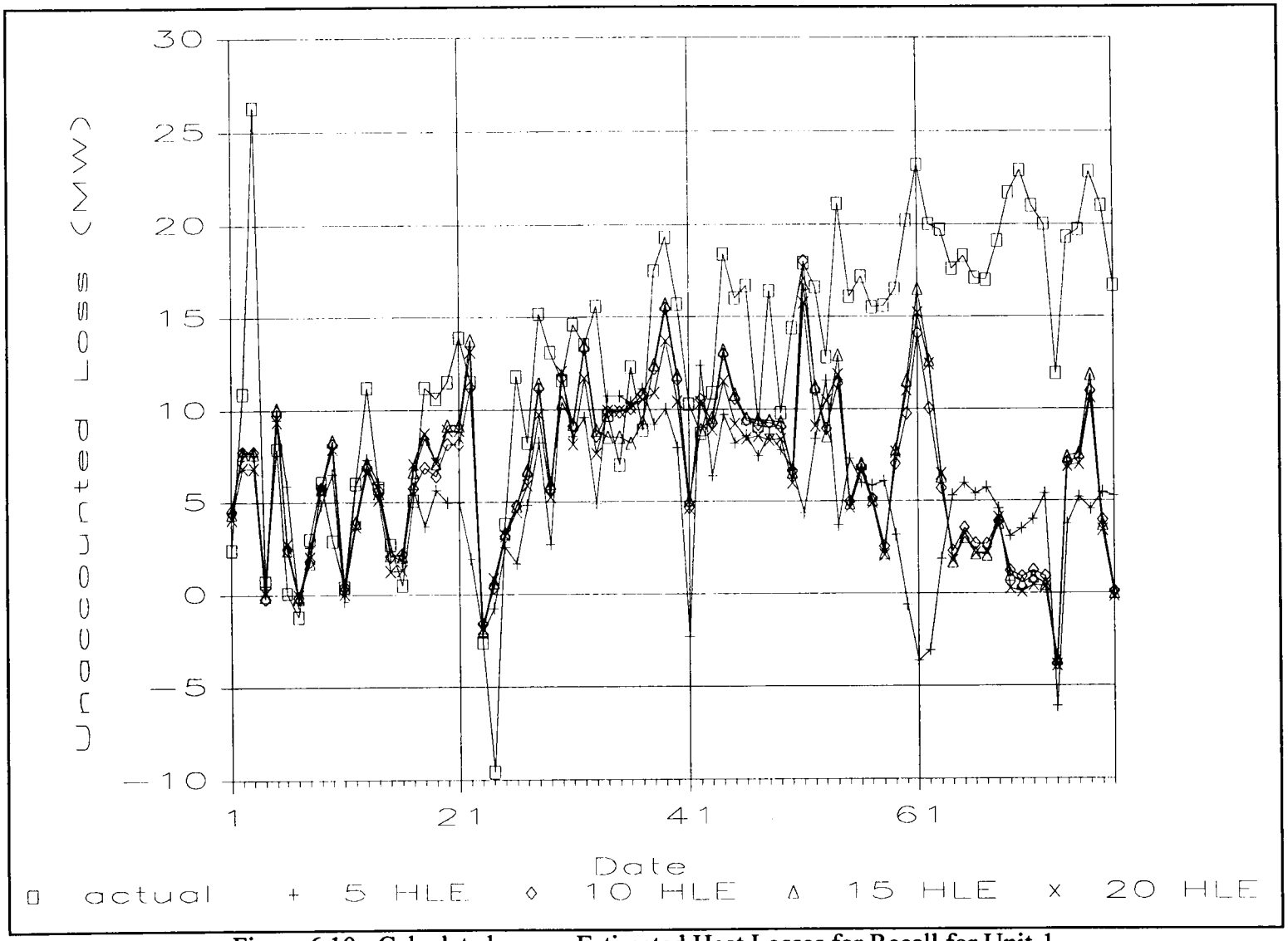


Figure 6.10: Calculated versus Estimated Heat Losses for Recall for Unit-1.

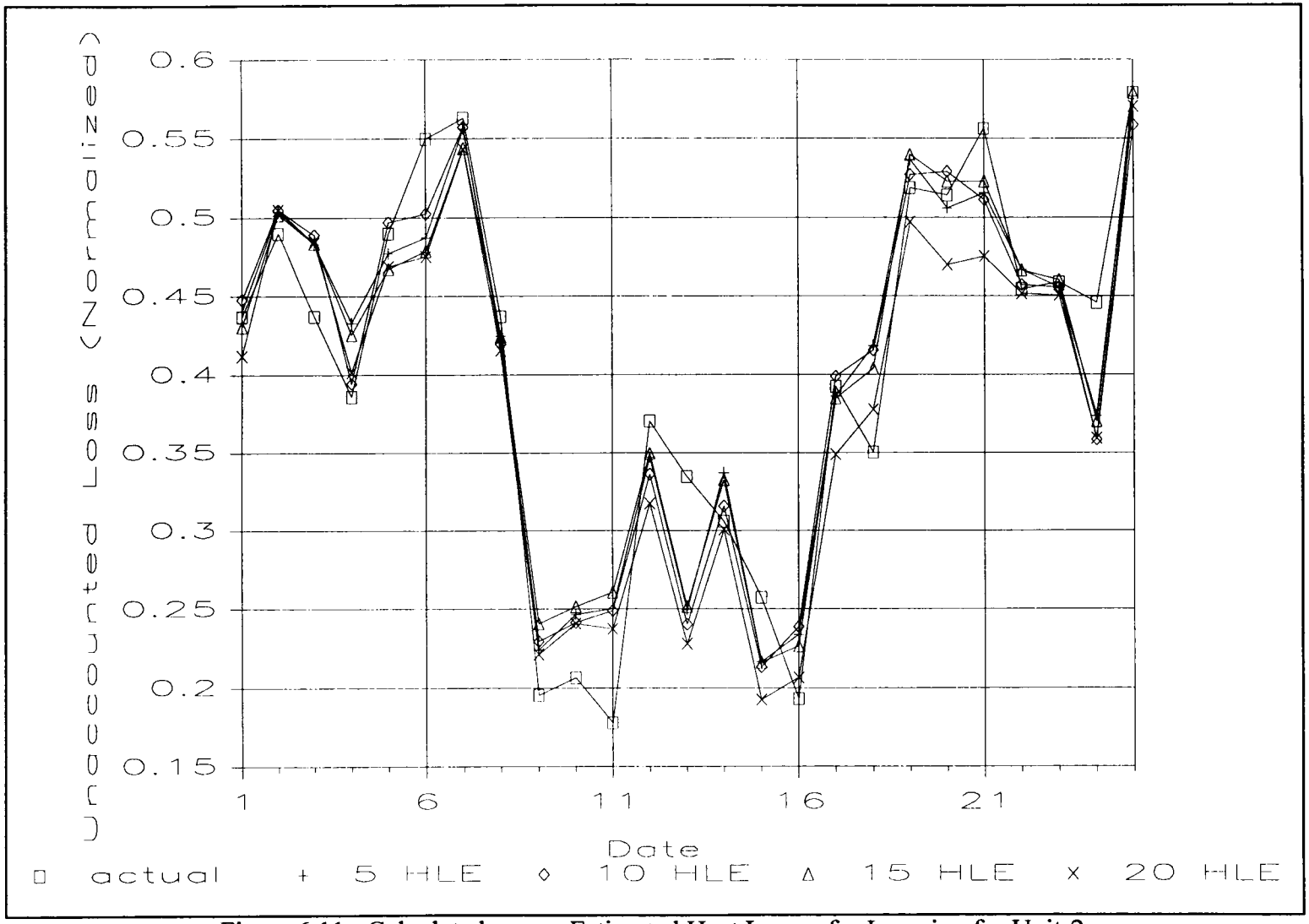


Figure 6.11: Calculated versus Estimated Heat Losses for Learning for Unit-2.

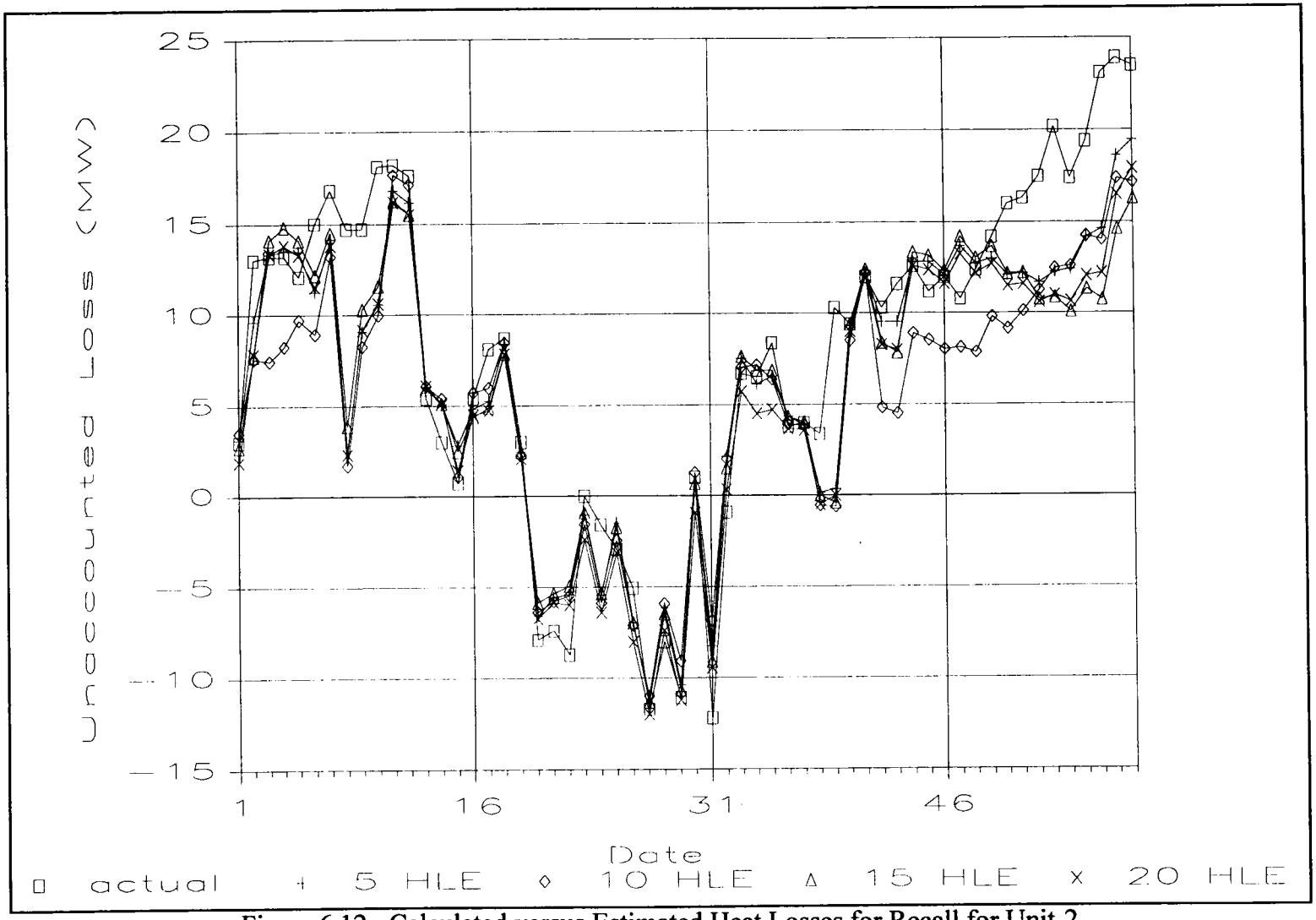


Figure 6.12: Calculated versus Estimated Heat Losses for Recall for Unit-2

Table 6.4: Learning Root-Mean-Square Errors.

# of HLPEs	UNIT-1 (MWe)	UNIT-2 (MWe)
5	0.251	0.200
10	0.245	0.202
15	0.242	0.202
20	0.249	0.214

Table 6.5: Recall Root-Mean-Square Errors.

# of HLPEs	UNIT-1 (MWe)	UNIT-2 (MWe)
5	3.24	1.92
10	3.01	2.14
15	2.99	2.06
20	3.05	2.05

HLPE stands for hidden layer processing elements.

A C computer code (see Appendix B) was written for calculating the heat rate of a typical Westinghouse-type PWR using the classical techniques [12]. There are approximately 40 input process variables to this code from all over the plant. Algebraic calculations based on classical heat and mass balance techniques are used to obtain the idealistic performance of the plant. Some empirical data, originally in the form of graphs (approximated by third order polynomials) were used for the non-idealistic corrections. IMSL mathematics package was used for evaluating the coefficients of these polynomials. The output of this code is the gross heat rate of the plant.

The data for this analysis were again taken from the weekly turbine cycle performance report data files of an operating four loop Westinghouse PWR. Fifteen signals (see Table 6.6) were selected to be the input signals to the network. All the signals in the data were normalized into the interval [0.2, 0.8]. There were a total of 41 patterns for the first unit (the patterns occur chronologically in the cycle) and with less unaccounted losses, were used as training patterns. The number of training patterns is 15 for this unit.

The network was trained up to a learning RMSE of 10^{-3} in the normalized domain and then used for recall. The results comparing the calculated and the estimated heat rates are given in Figure 6.13. The recall RMSE is found to be 3.78 Btu/kWh.

The results have shown a good agreement between the calculated and the estimated plant heat rate. Properly trained neural networks may be used in place of elaborate calculations, or these results may be used to cross-check plant calculations. Any deviation between the two must then be properly diagnosed.

Table 6.6: List of Signals for Heat Rate Estimation.

#	SIGNAL	UNIT
1	Feedwater Temperature	^{o}F
2	Main Steam to MSR A1	Psig
3	Impulse Pressure PT-1-73	Psia
4	Impulse Pressure PT-1-72	Psia
5	Average CCW Inlet Temperature	^{o}F
6	Condenser Backpressure	inches Hg
7	Steam Throttle Pressure	Psia
8	RCS Pressure	Psia
9	Cold Leg Temperature	^{o}F
10	Hot Leg Temperature	^{o}F
11	Vessel Blowdown Flow	GPM
12	Feedwater Pressure	Psia
13	Feedwater Flow Rate	lbm/hr
14	MSR Steam to LPT A Temperature	^{o}F
15	Steam Generator Pressure	Psia

where

MSR : moisture separator reheater

CCW: condenser cooling water

RCS : reactor coolant system

LPT : low pressure turbine

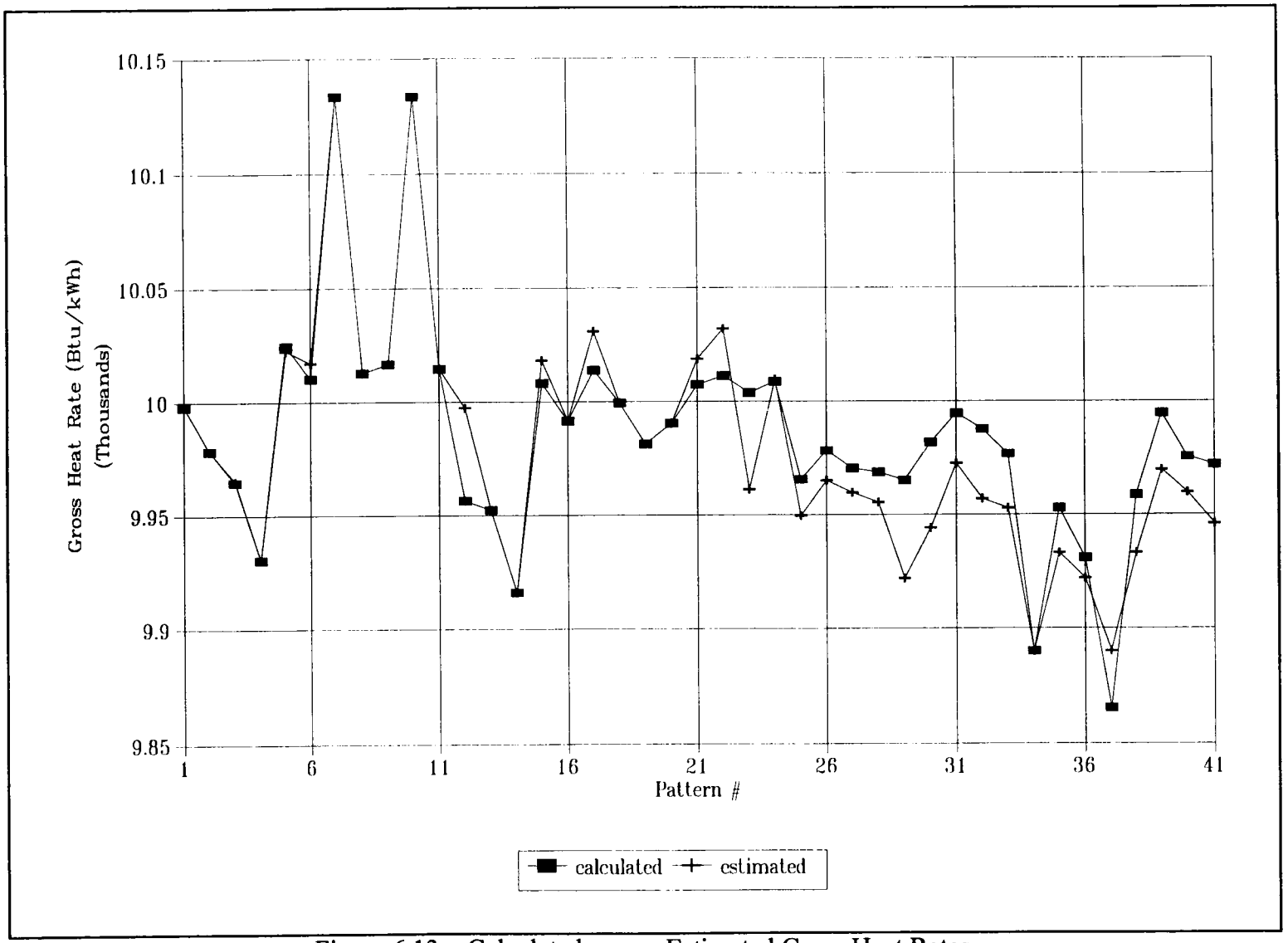


Figure 6.13: Calculated versus Estimated Gross Heat Rates.

6.5 Feedwater Flow Estimation

Feedwater flow rate is a very important measurement affecting the control strategy and thus efficient operation of nuclear power plants. Feedwater flow rate is one of the major signals in the determination of the thermal power of the plant. The classical measurement system using a venturi meter for this process variable suffers from a significant problem called fouling. This problem causes the measured flow rates to be artificially high which results in an artificially high thermal power. The present study uses ANN methodology to estimate the feedwater flow rate as a function of several other process variables in the system. Neural network technique was applied to two different types of PWRs. The following two subsections present and discuss the results obtained by using data from a 450 MWe PWR and two 1140 MWe PWRs.

6.5.1 A Four-Loop Pressurized Water Reactor

This part of the feedwater flow estimation analysis involves the use of operational data from two four-loop, 1140 MWe Westinghouse PWRs. Calculations are performed for each unit separately. A three-layer neural network was developed for each unit to estimate the feedwater flow rate as a function of seven variables. The input signals are given in Table 6.7, and were selected by considering heat and mass balances for the steam generators as explained in Chapter 2. The number of hidden layer nodes is 20 for the first unit and 16 for the second unit.

The data for this analysis were again taken from the weekly turbine cycle performance report data files of the two operating four loop Westinghouse PWRs. All the signals used

Table 6.7: List of Signals Used for Feedwater Flow Analysis of Four-Loop PWR.

#	SIGNAL	UNIT
1	Feedwater Temperature	^{o}F
2	RCS Pressure	Psia
3	Cold Leg Temperature	^{o}F
4	Hot Leg Temperature	^{o}F
5	Vessel Blowdown Flow	GPM
6	Feedwater Pressure	Psia
7	Steam Generator Pressure	Psia

in the network were normalized into the interval [0.2, 0.8]. There were a total of 41 patterns for the first unit and 22 for the second unit. For both units, the patterns close to the beginning of the cycle and with less unaccounted losses, were used as training patterns. The number of training patterns is 15 for the first unit, and 10 for the second unit.

Both networks were trained up to a learning RMSE of 10^{-3} in the normalized domain and then used for recall. The results comparing the measured and the estimated feedwater flow rates are given in Figures 6.14 and 6.15, for the first and the second unit of the plant, respectively. The learning and recall RMSEs are presented in Table 6.8.

Table 6.8: Learning and Recall RMSEs for Feedwater Estimations.

	LEARNING RMSE	RECALL RMSE	UNIT
UNIT-1	7.6026	66.0541	lbm/h
UNIT-2	4.8156	91.8758	lbm/h

While there is no significant difference between the measured and the estimated feedwater flow rates for the first unit, the measured rates are generally higher than the estimated values for the second unit. However, the trend is so complex that it does not yield any clear conclusions.

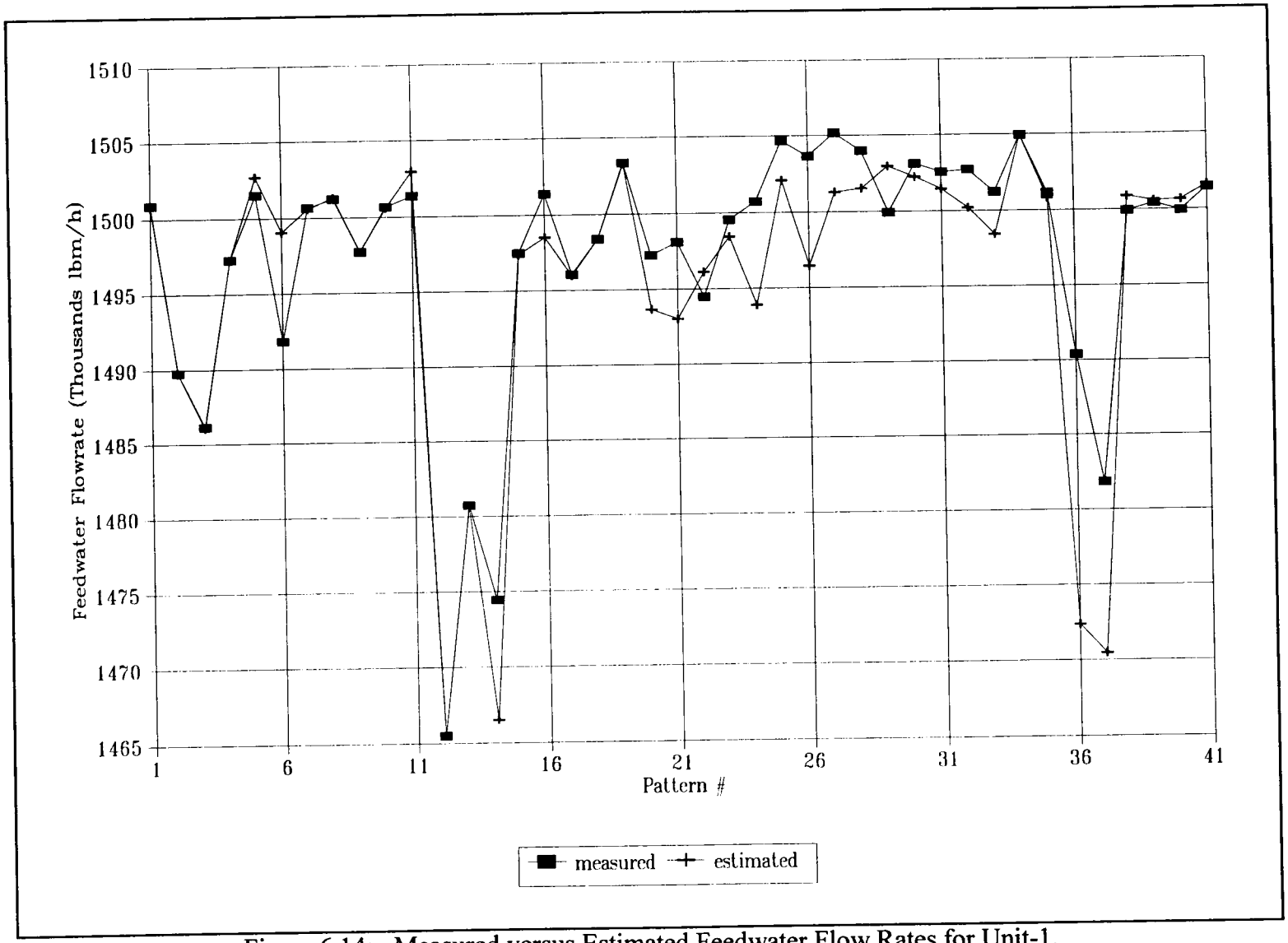


Figure 6.14: Measured versus Estimated Feedwater Flow Rates for Unit-1.

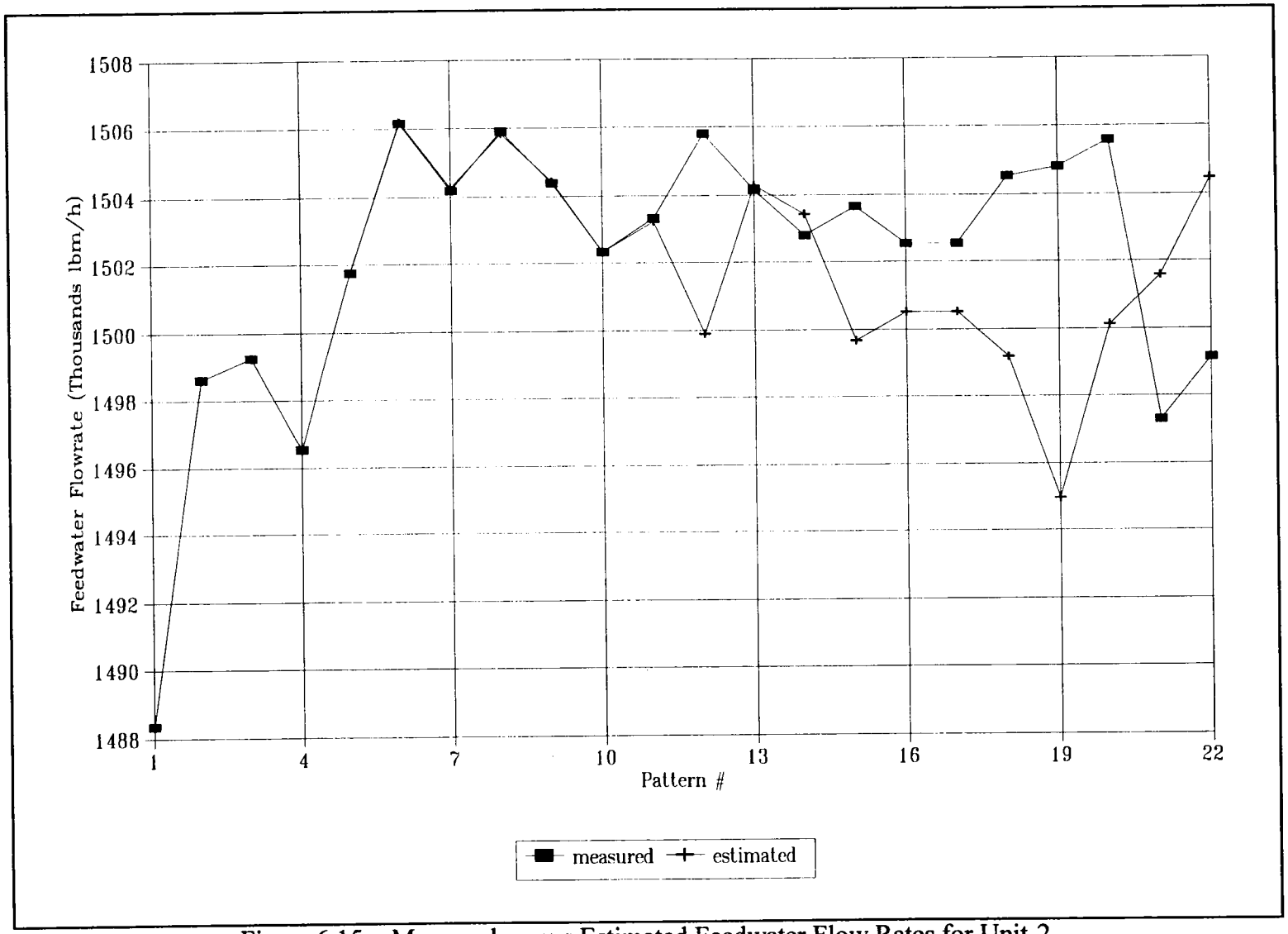


Figure 6.15: Measured versus Estimated Feedwater Flow Rates for Unit-2.

6.5.2 A Six-Loop Pressurized Water Reactor

The data for this analysis were received from a nuclear power plant which is a 450 MWe, six-loop pressurized water reactor. The data file consists of the daily averages of 141 various signals for the period of one year. For each steam generator nine signals were extracted and are given in Table 6.9 from the original data file for the whole year.

Because of the nonlinear transfer functions (such as the sigmoid) used in neural networks, the input and the output signals to network are normalized into a small interval in order to avoid network saturation. If the data is not distributed uniformly over the maximum and the minimum of the related signal, the normalization process destroys the performance of the network.

For the above reason, these nine signals for each loop were visually analyzed first. The patterns (the days) which exhibited some form of abnormality, such as a valley or a peak which may correspond to a change in power operation, were removed from the data file. Almost all the patterns removed were from the last three months of the original data set.

All the nine signals extracted from the original data file are listed in Table 6.9. For the studies reported here, the first eight signals in Table 6.9 were used as input to the models for each loop. All the analyses were carried out for each loop separately.

Table 6.9: List of Signals Used for Feedwater Flow Analysis of a Six-Loop PWR.

#	SIGNAL	UNIT
1	Hot Leg Temperature	^{o}C
2	Cold Leg Temperature	^{o}C
3	Feedwater Temperature	^{o}C
4	Reactor Pressure Drop	bar
5	Feedwater Pressure	bar
6	Steam Pressure	bar
7	Reactor Coolant Flow Rate	kg/s
8	Vessel Blowdown Flow Rate	kg/s
9	Feedwater Flow Rate	kg/s

6.5.2.1 Least-squares Estimation of Flow Rate

Generally, a heat and mass balance analysis on steam generators is performed to determine feedwater flow rate in power plants. Since there is no analytical way of calculating the actual value of the overall effect of pump work and the various heat losses, most plants use a nominal value for the overall effect to utilize the heat balance to obtain flow rate. In the present analysis we performed a least-squares (LS) fit for determining the relationship between the secondary and the primary thermal powers. This relation was used to determine the feedwater flow rate.

Since we restricted our study to full power operation, the relationship between primary and secondary powers is probably a shift in the magnitude only, i.e. the following form is used.

$$Q_{sec} = Q_{pri} + \alpha \qquad (6.2)$$

where α is the net effect of various heat losses and the pump work between the primary and the secondary system.

α was calculated using a single least-squares fit. Equation (6.2) gives the following relation for the feedwater flow rate.

$$w_{fw} = w_{RCS}\frac{h_{HL} - h_{CL}}{h_s - h_{fw}} + w_{vbd}\frac{h_s - h_{vbd}}{h_s - h_{fw}} + \frac{\alpha}{h_s - h_{fw}} \qquad (6.3)$$

6.5.2.2 Neural Network Estimation

The goal of this study is to isolate the feedwater flow rate signal and to estimate the flow rate in order to monitor possible fouling of venturi tube. This anomaly results in artificially high flow rates. Therefore, hetero-associative neural networks were developed to predict the feedwater flow rate for each loop of the plant. If a network is trained using the data acquired early in the fuel cycle (when the venturi tube is clean), it is expected to predict the correct flow rates in the future, since the network represents a correct model of the feedwater flow rate as a function of other variables.

For each loop of the plant, a neural network was built with eight input and one output layer processing elements. The inputs are the first eight signals in Table 6.9. The output processing element is the feedwater flow rate. As stated earlier, since it was believed that the data for the first two weeks were accurate enough, the first 14 patterns were used to train the neural networks for the six loops.

All the networks used in this study are three layer networks with 12 processing elements in their single hidden layer. Starting from a high value for the hidden layer elements, this number was decreased until the same learning root-mean-square error was established. Then this number was used for all networks. The interval [0.2 0.8] was used as the normalization interval. The learning and momentum coefficients were given high values at the beginning of learning and were decreased as the learning proceeds. The sigmoid was used as the transfer function.

The neural networks were trained to give a root-mean-square error of 10^{-2} in the normalized domain. The trained networks were used to estimate the feedwater flow rate for the remaining data.

There are two estimates of the feedwater flow rate; one from the least-squares method and the other from the neural networks for each of the six loops. The results for the first and the second loops are given in Figures 6.16 and 6.17. Table 6.10 summarizes the learning and overall root-mean-square error between the estimation and the measured feedwater flow rate.

Table 6.10: Learning (L) and Recall (R) RMSEs for Neural Network (NN) and Least-Squares (LS) Estimations.

	LOOP-1	LOOP-2	UNIT
L-NN	0.0955	0.0634	kg/s
L-LS	0.1786	0.3379	kg/s
R-NN	0.6677	0.3652	kg/s
R-LS	1.0437	0.4885	kg/s

Estimation analysis was performed for each of the six loops, but the results are presented for loops 1 and 2. The results for the first loop are similar to the results for the third and the fifth loops, and the results for the second loop are similar to the results for the fourth and the sixth loops.

The estimations of feedwater flow rate by both the neural networks and the least-squares fits are lower than the measured flow rates for loops one, three, and five; whereas the estimates roughly agree with the measurements for loops two, four and six.

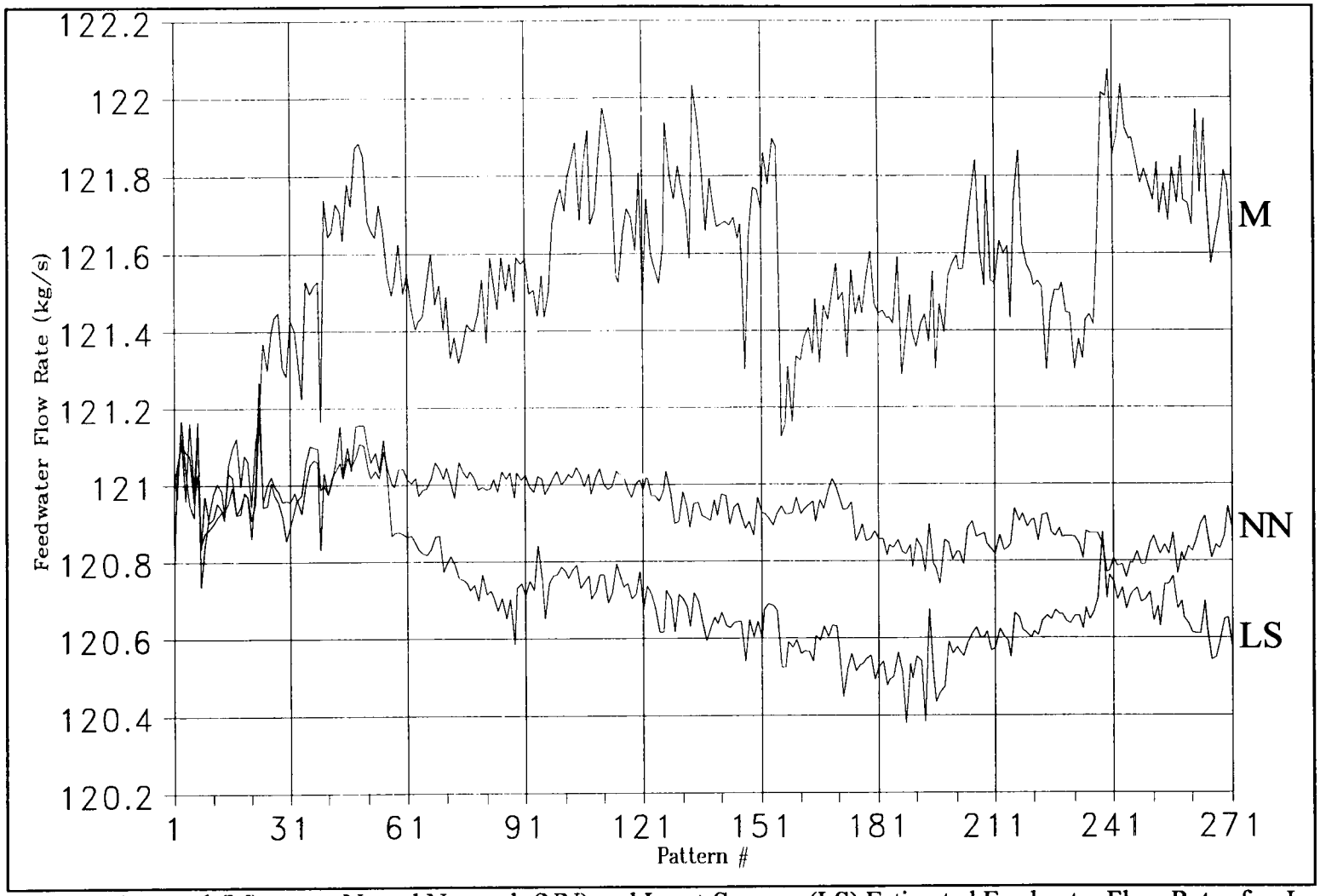


Figure 6.16: Measured (M) versus Neural Network (NN) and Least-Squares (LS) Estimated Feedwater Flow Rates for Loop-1 .

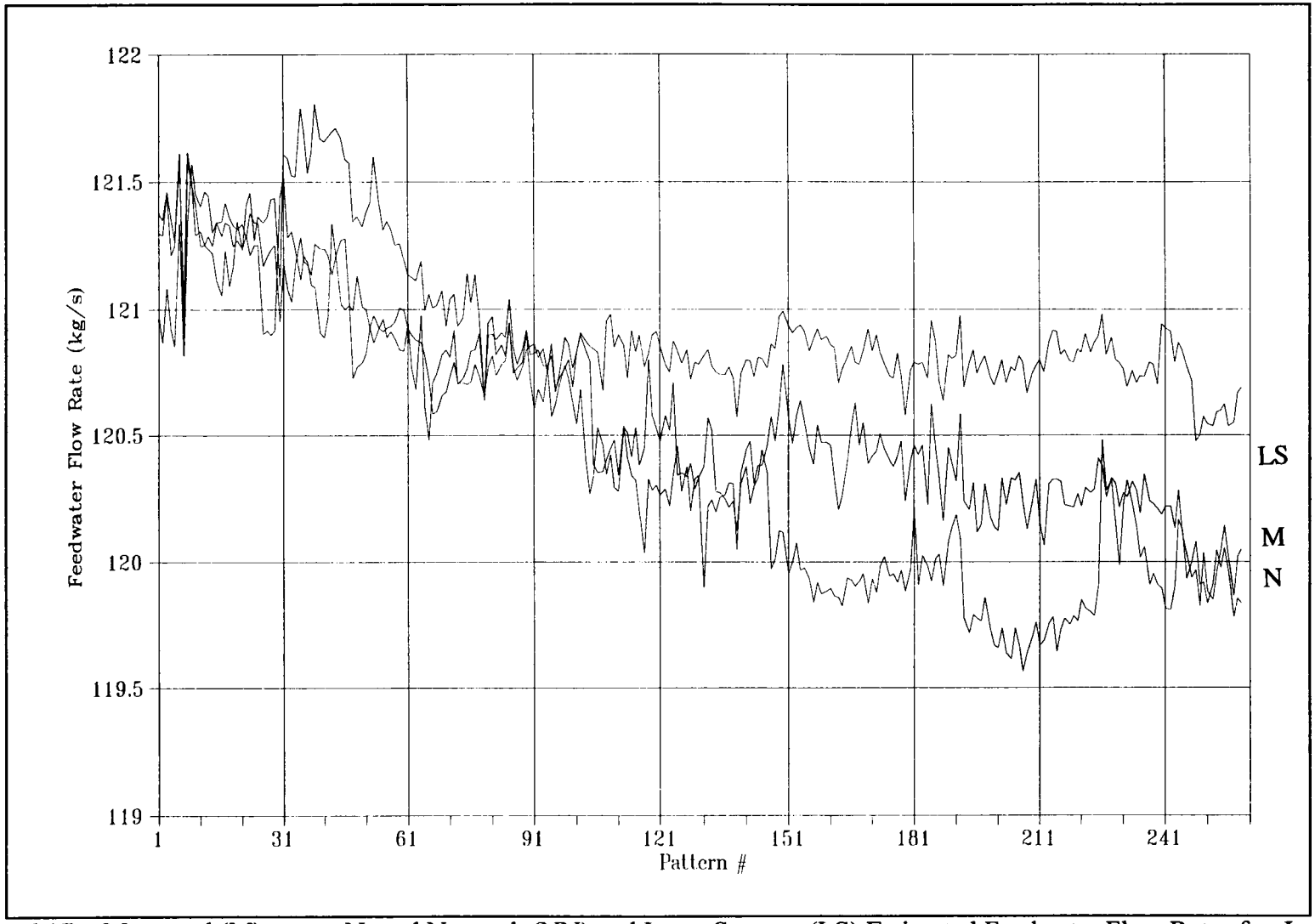


Figure 6.17: Measured (M) versus Neural Network (NN) and Least-Squares (LS) Estimated Feedwater Flow Rates for Loop-2.

The reason for the deviations seen in the estimations for loops one, three, and five may be due to venturi fouling or some other anomaly. A change in the system such as a pump or valve might have changed the interrelationship among the signals. The use of erroneous learning patterns might have also been a reason for the difference in the recall phase.

6.6 Sensitivity Analysis

The sensitivity of an input-output model may be used to determine the most important input signals contributing to the output. Sensitivity analysis using neural network models was presented in Chapter 5. Feedwater flow estimator networks for the four-loop Westinghouse-type PWR (see section 6.4.1) were used as network models for the sensitivity analysis. An arbitrary training pattern was selected for each network for the first and the second units of the plant.

First, the differential sensitivities were evaluated experimentally without using the formula derived earlier. The following fact is used for this purpose.

$$S_x^y = \frac{\partial y}{\partial x} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \qquad (6.4)$$

For each of the seven signals given in Table 6.6, the input signal is perturbed in a small amount approaching to zero, and the resulting output deviation was found by using network recall. The output is the feedwater flow rate. For hot leg temperature as the input, the ratio $\Delta y/\Delta x$ versus Δx is given in Figure 6.18 for the two units

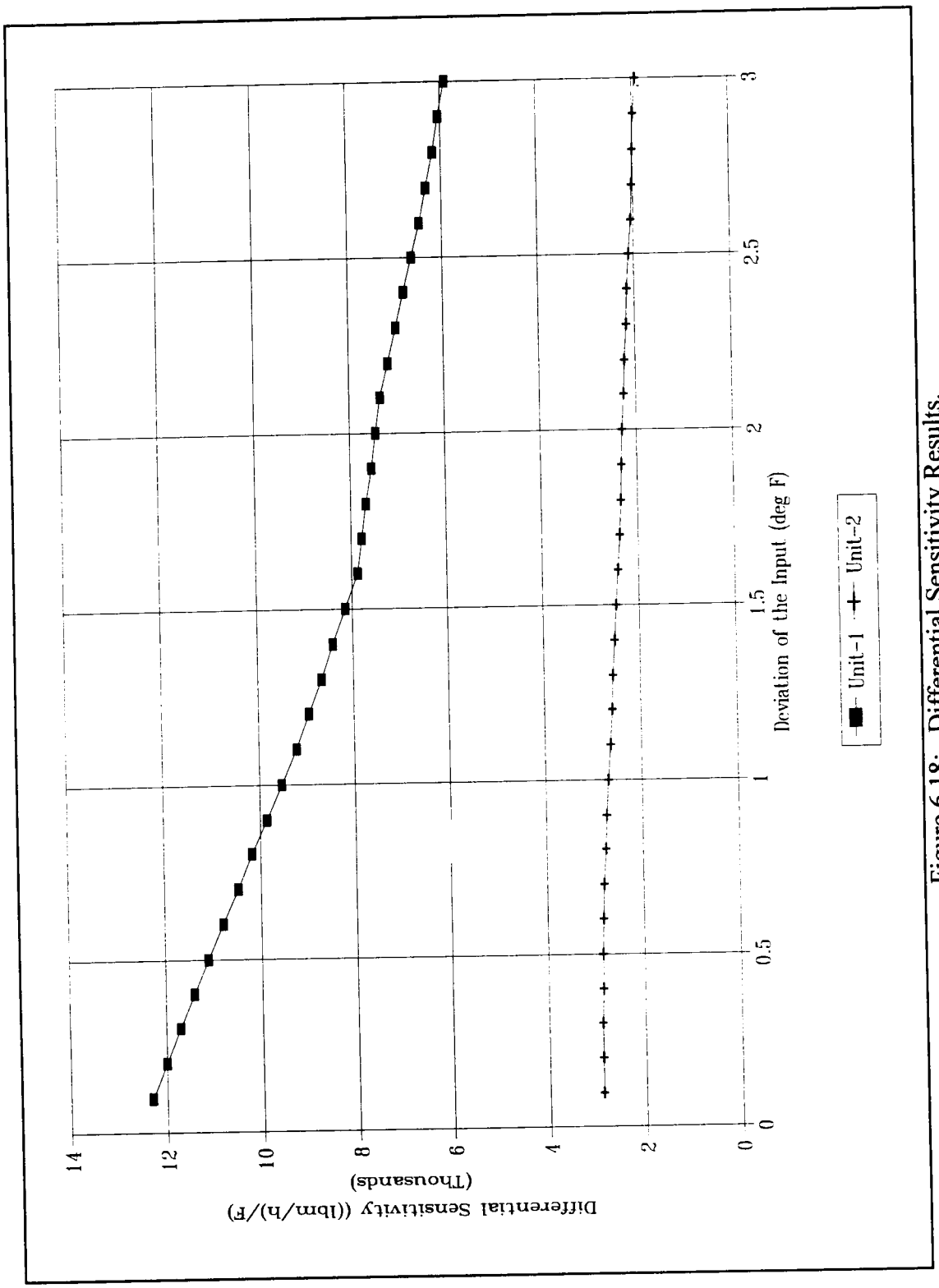


Figure 6.18: Differential Sensitivity Results.

Secondly, an experimental evaluation of the statistical sensitivity of the output to any of the inputs was performed. For each of the inputs, ten different (different standard deviations) Gaussian sequences of 100 elements with mean zero were created, and added to the input signal. These perturbed inputs were presented to the network and the output deviations were evaluated. For hot leg temperature as the input, the ratio of output standard deviation versus the input standard deviation is given in Figure 6.19 for the two units. Table 6.11 presents the order of importance of input signals according to the neural network models.

Results have shown that different sensitivity measures yield similar results for the output sensitivity. On the other hand the sensitivities of feedwater flow to inputs were not very close for different units. However, for both units, hot leg temperature is found to be the most important signal, whereas the vessel blowdown flow is the least important. In general, temperatures are more important than the pressures for the flow rate. A final point that must be made is that all the sensitivities depend on the input pattern. Any deviation in these patterns will affect the order of importance of signals.

6.7 Thermal Performance Diagnostics

The detection of degradation in a component of the power plant is important for isolating the causes of a degradation in overall plant performance, such as the heat rate. Properly trained neural networks can detect the changes in the behavior of individual systems. A feedwater heater was chosen to be the sample component for this analysis. A **C** code was written to estimate the efficiency of the component (see section 2.3) based on the first law of thermodynamics and the steam tables. A single-output three-layer network was built to

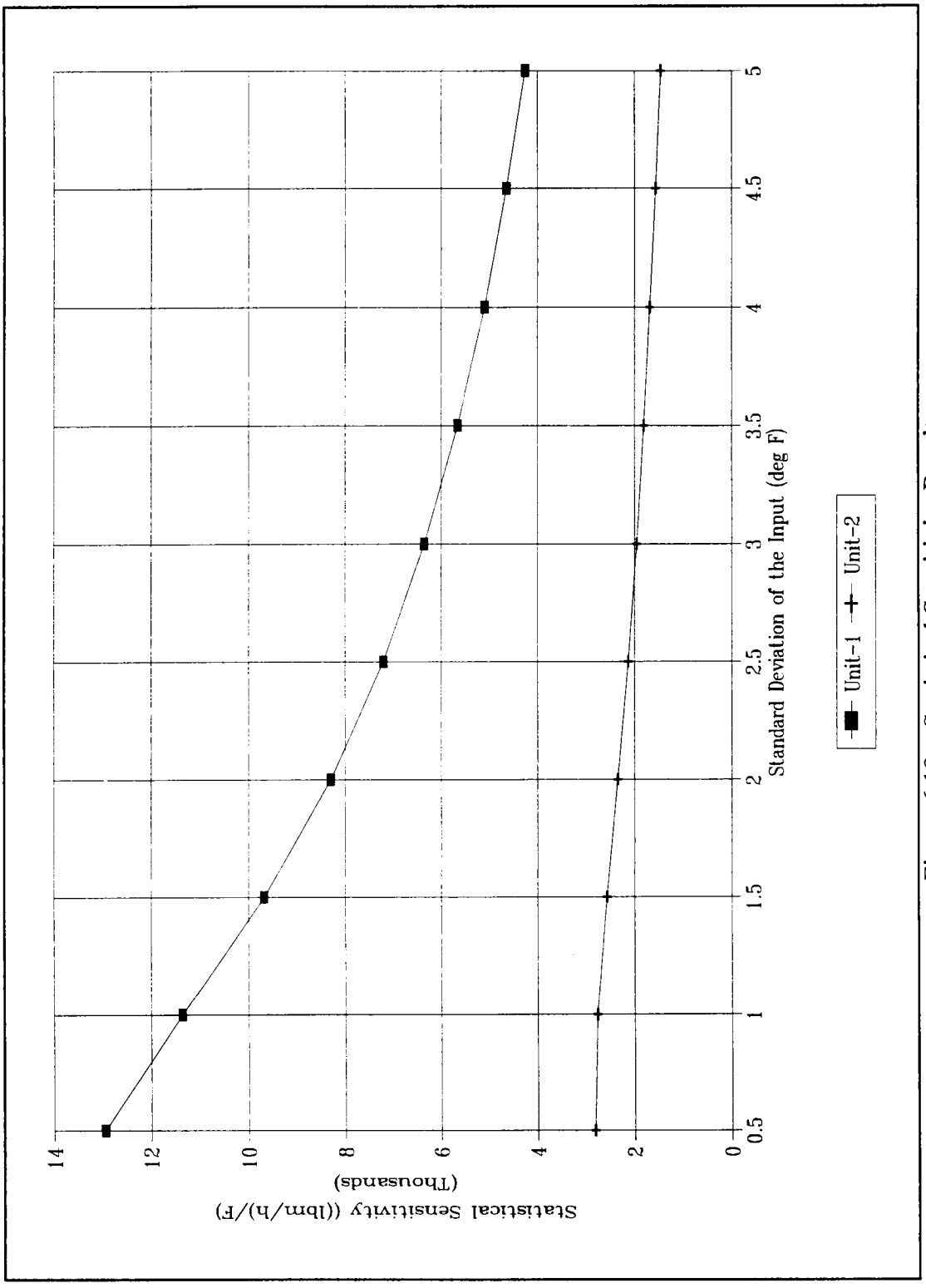


Figure 6.19: Statistical Sensitivity Results.

Table 6.11: Order of Importance of Inputs for Feedwater Flow Rate.

ORDER	UNIT-1	UNIT-2
1	Hot Leg Temperature	Hot Leg Temperature
2	Cold Leg Temperature	Feedwater Temperature
3	Feedwater Temperature	RCS Pressure
4	Steam Generator Pressure	Cold Leg Temperature
5	RCS Pressure	Feedwater Pressure
6	Feedwater Pressure	Steam Generator Pressure
7	Vessel Blowdown Flow	Vessel Blowdown Flow

estimate the efficiency of the component. Twelve signals (see Table 6.12) that are necessary to determine the thermal efficiency of the heater (see Section 2.3), are used as the input signals for this mapping. The number of hidden layer processing elements is 12.

The data for this analysis were hourly averages of the archived data of the 12 signals. There were a total of 110 patterns, and 30 of them were used as training patterns. The efficiency was first calculated analytically. These values were used to train the network. The normalization interval was [0.2, 0.8].

The network was trained up to a learning RMSE of 10^{-3} in the normalized domain, and used for recall. The results are presented in Figure 6.20. The results have shown the feasibility of applying neural networks for thermal performance diagnostics purposes. This specific example indicates that a change in the performance of the feedwater heater may be detected by a neural network (in this case a small decrease in the efficiency).

6.8 On-line Implementation

Feedwater flow monitoring and heat rate monitoring parts of this study are being implemented in an active environment on a SUN Sparc workstation as a plant monitoring and diagnostics module. A computer code reads all the input information for the estimation networks and calculates the feedwater flow rate and the heat rate. The on-line plots for these variables are updated using the results of current calculations. Figures 6.21 and 6.22 show the main menu and the monitoring screen of this code.

Table 6.12: List of Signals Used for Diagnostics.

#	SIGNAL	UNIT
1	FWH 4A Steam Inlet Pressure	Psia
2	FWH 4B Steam Inlet Pressure	Psia
3	FWH 4C Steam Inlet Pressure	Psia
4	FWH 4A Drain Temperature	°F
5	FWH 4B Drain Temperature	°F
6	FWH 4C Drain Temperature	°F
7	FWH 4A Feedwater Inlet Temperature	°F
8	FWH 4B Feedwater Inlet Temperature	°F
9	FWH 4C Feedwater Inlet Temperature	°F
10	FWH 4A Feedwater Outlet Temperature	°F
11	FWH 4B Feedwater Outlet Temperature	°F
12	FWH 4C Feedwater Outlet Temperature	°F

where FWH stands for feedwater heater.

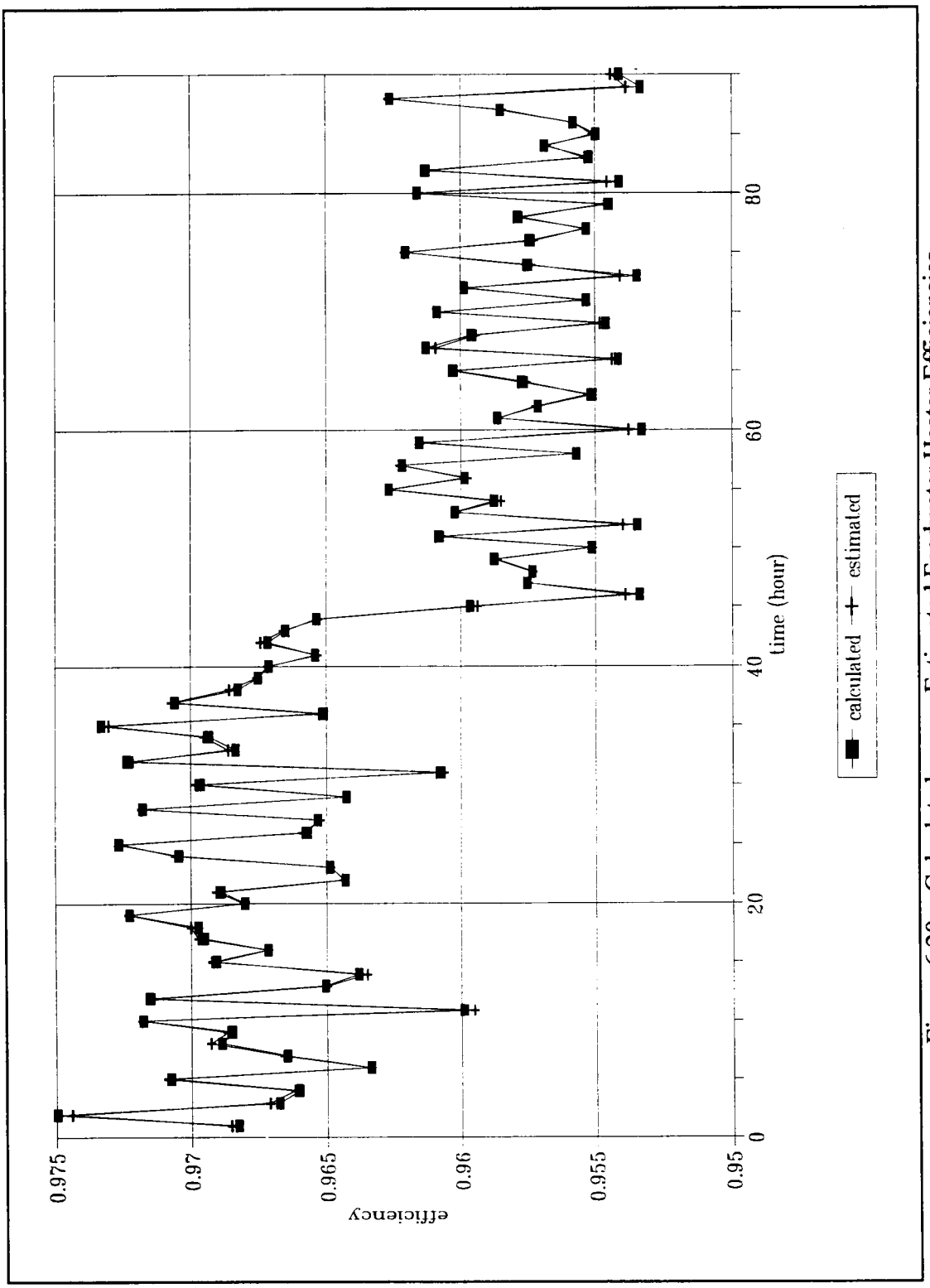


Figure 6.20: Calculated versus Estimated Feedwater Heater Efficiencies.

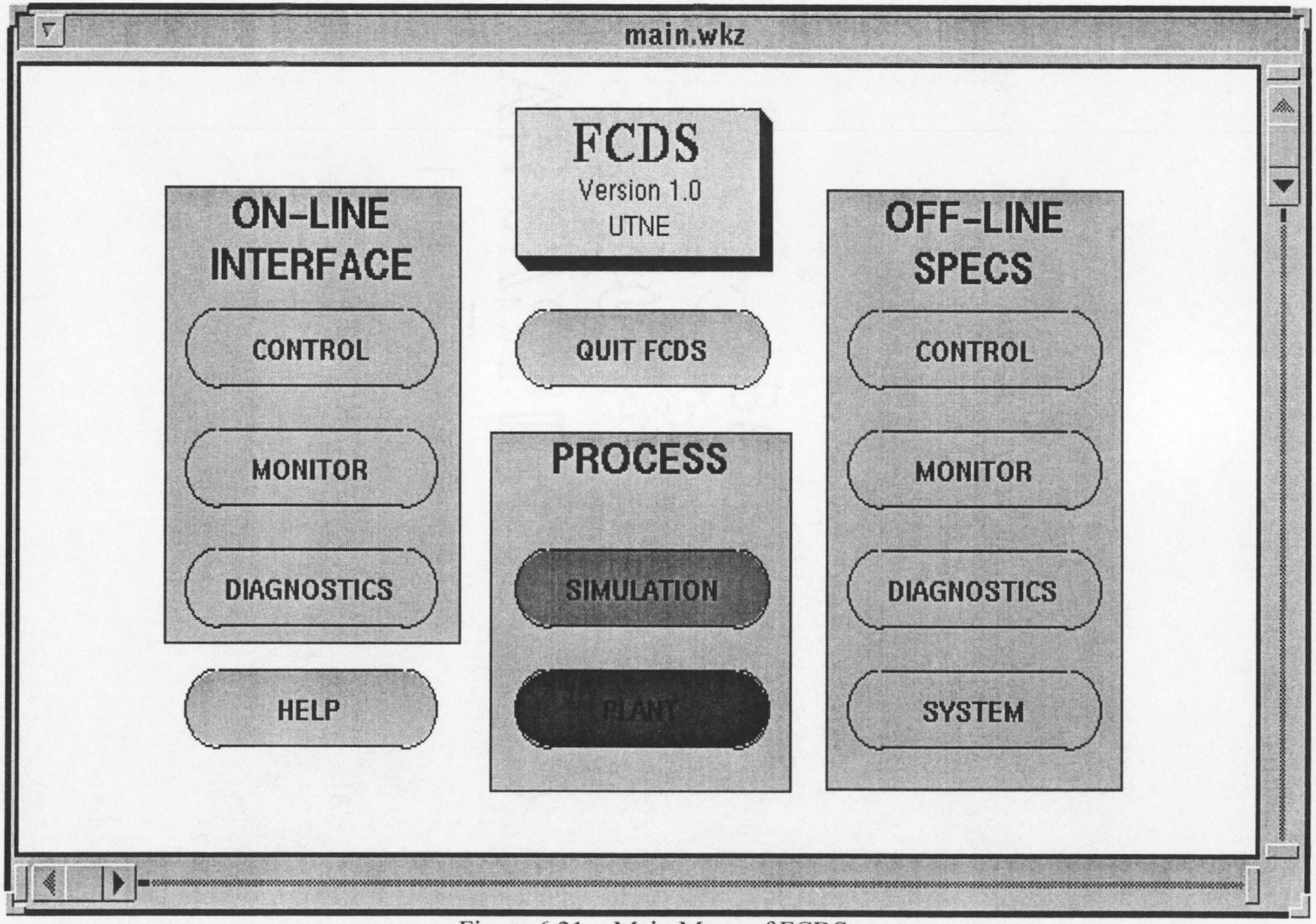


Figure 6.21: Main Menu of FCDS.

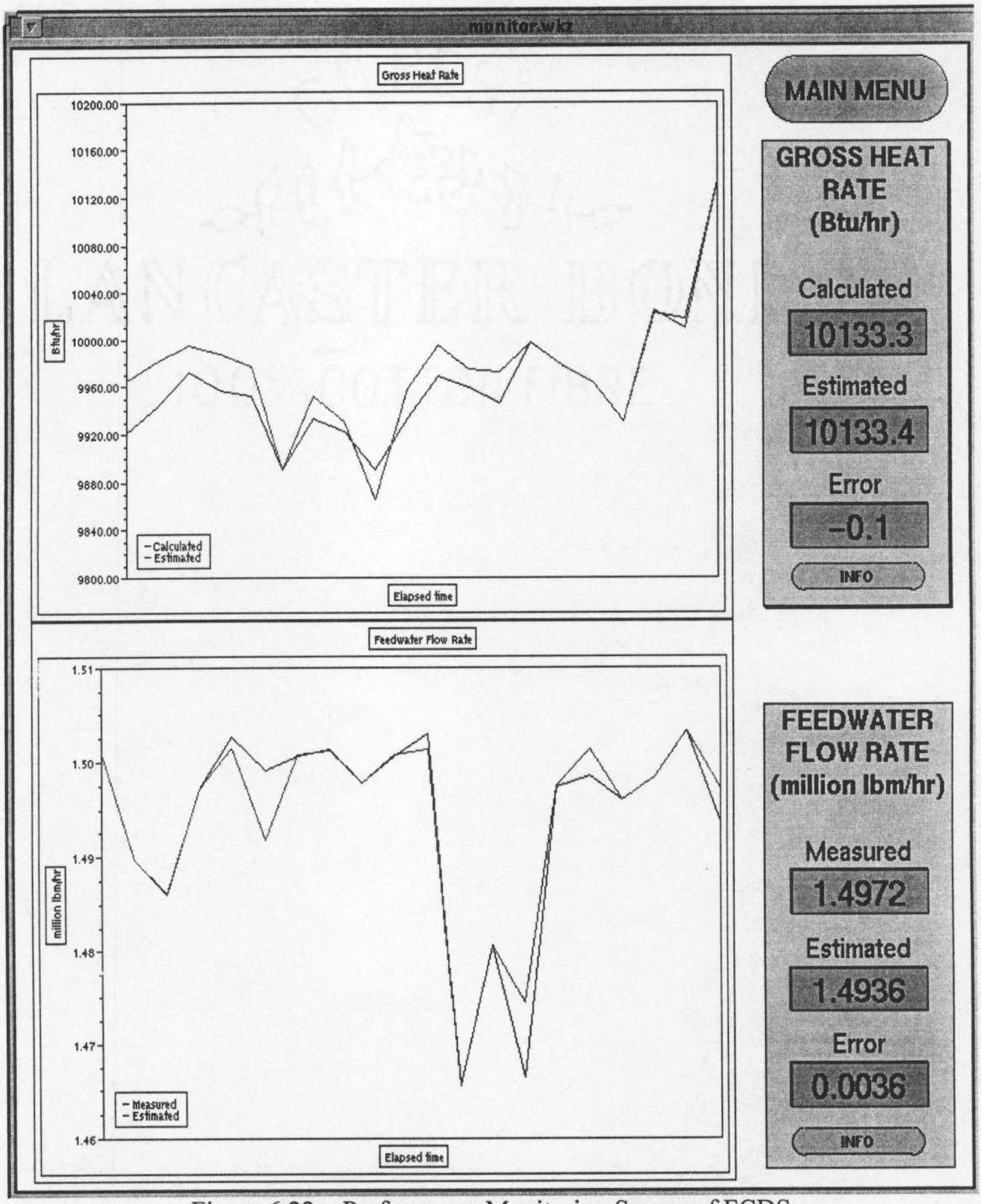


Figure 6.22: Performance Monitoring Screen of FCDS.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

7.1 Summary

The multi-layer perceptron that uses backpropagation learning algorithm was applied to several areas of nuclear power plant monitoring and diagnostics. Keeping the plant heat rate as low as possible and maintaining full power operation are two of the most important operational objectives for the power plant industry.

Various signals from both the primary and the secondary sides of a nuclear power plant were monitored using auto-associative networks to keep track of any changes in the system behavior. Auto-associative backpropagation networks were used and found to be very successful in this application.

The unknown portion of the plant heat losses was estimated as a function of selected process variables in the system using neural networks. Different hidden layer numbers were tried and a comparison is made between the results form different networks. The patterns where the unaccounted heat losses are less than some acceptable value are used as training patterns for the networks.

Properly developed artificial neural networks were implemented to estimate feedwater flow rates in a four-loop and a six-loop PWR plants. Actual data from two different types

of operating pressurized water reactors were used for this analysis. The patterns recorded early in the fuel cycle were used as training patterns, since the venturi is supposed to be clean during that period, and consequently the feedwater flow rate measurements are accurate early in the cycle.

A sensitivity analysis was performed on the neural network models for estimating the feedwater flow rate in order to find the most important input signals contributing to this flow rate. Both the differential and the statistical sensitivity measures were used to determine the model's output sensitivity. An arbitrary pattern from the training set was selected as the nominal input to the networks.

A computer code that calculates the plant heat rate based on the currently used thermodynamic analysis techniques was developed. This calculation uses 40 input signals. Various process variables from the whole system were used as input signals to a network whose single output is the heat rate of the plant. It should be noted that the input signals are all measurable process variables, not the calculated parameters.

A method for monitoring the thermal performance of individual components of a system was proposed. An efficiency measure was defined for that component, and a computer code was generated that calculates the thermal efficiency of the heater using steam tables. Thermodynamic and neural network models were compared based on the results. This approach is very useful in detecting anomalies in one or more components of a large system whose overall performance would be affected by sub-system performance.

Feedwater flow rate estimation and heat rate estimation parts of this thesis are being implemented in a SUN Sparc workstation as an on-line power plant monitoring tool. A computer code reads all the necessary input information and performs the neural network estimations of heat rate and feedwater flow rate using previously trained networks. The results are used to update the plots on-line for the related variables.

7.2 Conclusions

In general, the results obtained from the studies in this thesis have shown the feasibility of applying artificial neural networks to nuclear power plant performance monitoring and diagnostics problems. The secondary side of PWRs is the focus of this study. There are some advantages of using neural network models compared to the classical methods. There are also some drawbacks in this approach.

It was found that various signals from a power plant can be monitored in an auto-associative manner without knowing the physical relationship among them. Auto-associative monitoring provides a way of diagnosing the abnormalities in the system or some of the process variables. Neural networks offer a simple way of doing this analysis without performing all the elaborate classical calculations.

Properly trained networks can also be used for estimating the unaccounted heat losses in a plant and is an indicator of the plant's thermal performance. These networks can also be used to monitor long-term trends in the heat losses.

Another important conclusion of this thesis is that the feedwater flow rate to the steam generators can be monitored using artificial neural networks. These network models may also be used to detect the fouling of feedwater venturi meter, which introduces artificially high flow rates and therefore artificially high plant thermal powers. However, accurate quantification of the effect of venturi fouling may not be possible by neural networks. The reason for this could be the possibility of using slightly fouled venturi meter readings as training patterns for the networks.

The sensitivity analysis based on neural network models has shown that the most important signal that contributes to feedwater flow rate is the hot leg temperature, and the least important signal is the vessel blowdown flow. However, it should be emphasized that these sensitivities depend not only on the network connection weights, but also on the input pattern used for evaluating them. Therefore, there may be slight changes in the order of importance of input variables by using different input patterns.

It is also concluded that the plant heat rate can be estimated using neural networks with very good accuracy. The results have shown that the difference between the analytically calculated and the neural network estimated heat rates is very small. Neural network models, in general, require less number of input signals and less computational time than the classical methods.

The results have also shown that neural networks can be successfully applied to diagnose the degradations in the thermal performance of components of a power plant. Small changes in the thermal efficiency of a feedwater heater were detected precisely by a network. The methodology can be extended to the whole power plant. If a neural

network is developed for each component of the system whose single output is a measure of the thermal efficiency of the related component, then, by combining these networks, the cause of an overall plant heat rate degradation problem can be localized to the component that is the source the problem.

On the other hand, there are some drawbacks in the application of neural networks to power plant thermal performance problems. Data pre-processing was found to have a very important role in the performance of neural network models. The ranges that input and the output signals can vary should be determined very carefully. The normalization interval should be well suited to the specific application.

Since there is no obligation to include or exclude a process variable as a model input for neural networks in contrast to the physical models, determining the input signals for estimating a given variable depends on the analyst. Using very few inputs may result in network's inability to perform the desired mapping, whereas using too many signals as input may introduce unnecessary complexities and erroneous results.

The determination of network parameters is still an open question. Selecting the number of hidden layer elements, choosing a transfer function, and determining a learning schedule, along with learning and momentum coefficients, should be done by the analyst by a trial-and-error approach.

Another disadvantage of neural network models is the need for updating the model, especially for on-line applications when there is a design change, component replacement

or some similar change in the power plant. Neural networks are sample-based models and the training samples must cover the dynamic range over which recall occurs.

Neural networks also cannot provide a physical insight into the problem unless they are designed to function so. In other words, they cannot generate reasons for producing a certain output. Therefore, the analyst should develop the methodology such that it allows diagnostics types of conclusions based on the results.

7.3 Recommendations for Future Research

The analyses presented throughout this thesis are planned to be extended using data directly from plant process computers instead of using data files of weekly turbine cycle performance tests. Database management issues must be resolved in on-line applications.

Feedwater flow estimator networks were trained using daily or weekly data from the power plants that cover the period of about two weeks. Considering that the venturi fouling process is completed approximately in two months, and it starts from the beginning of the fuel cycle, the future work must utilize more frequently sampled data for training when the venturi is very clean.

Another extension to this study is to monitor the trends in some other process variables such as turbine impulse pressure and reactor core differential temperature that are closely related to power plant thermal performance. In addition, a systematic approach for monitoring thermal efficiencies of components on the secondary side is also desirable.

The effects of instrument inaccuracies on the estimated quantities should also be analyzed.

Finally, testing and verification of the neural network models created throughout this thesis will be performed in an operating nuclear power plant as an off-line monitoring and diagnostics module.

LIST OF REFERENCES

LIST OF REFERENCES

[1] K. Booth, "On-line Monitoring Energizes Power Plant Performance," INTECH Applying Technology, p 44, October 1991.

[2] "Fluid Meters, Their Theory and Application," ASME Report, Research Committee on Fluid Meters, p 43, New York, 1971.

[3] E. O. Doebelin, " Measurement Systems Applications and Design," McGraw-Hill, p 528, New York, 1983.

[4] Ontario Hydro, "Feedwater Flow Measurement in U. S. Nuclear Power Generation Stations," EPRI TR-101388, November 1992.

[5] Nuclear News, "Flowrate Measurement Causes Unneeded Derating," vol. 36, no. 2, p 39, February 1993.

[6] K. Kavaklioglu, B. R. Upadhyaya and E. Eryurek, "Neural Networks for Feedwater Flow Estimation in PWRs," Proceedings of 8th Power Plant Dynamics Control & Testing Symposium, Knoxville, Tennessee, vol. 2, p 87.01, May, 1992.

[7] P. K. Simpson, "Artificial Neural Systems," Pergamon Press, New York, 1990.

[8] R. Hecht-Nielsen, "Neurocomputing," Addison-Wesley, Reading, Massachusetts, 1990.

[9] "Technical Instruction TI-2.1," Calorimetric Calculation for Unit-1, Tennessee Valley Authority, Sequoyah Nuclear Power Plant, June 1990.

[10] J. E. Mott and P. Blanch, "Feedwater Flow Estimation via Sample Based Modeling," 8th Power Plant Dynamics Control & Testing Symposium, Knoxville, Tennessee, vol. 2, p 85.01, May, 1992.

[11] M. Khadem, F. J. Alexandra, and R. W. Colley, "Sensor Validation in Power Plants Using Neural Networks," EPRI, International Neural Network Society, Summer Workshop on Neural Network Computing for the Electric Power Industry, Stanford University, August, 1992.

[12] "Periodic Instruction 0-PI-SXX-000-001.0," Turbine Cycle Performance Test, Tennessee Valley Authority, February 1990.

[13] E. Interess and S. B. Sabol, "A Methodology to Minimize Unaccountable Heat Rate Losses," Proceedings of EPRI Heat Rate Improvement Conference, GS-6635, p 4A-1, January 1990.

[14] J. A. Babb, "Heat Rate Improvement : Theory and Application at Southern California Edison." Proc. of EPRI Heat Rate Improvement Conference, GS-6635, p 3A-33, January 1990.

[15] P. S. Lance, "An Integrated Performance Monitoring System for Fossil Power Plants with On-line and Off-line Applications." Proc. of EPRI Heat Rate Improvement Conference, GS-6635, p 4A-87, January 1990.

[16] J. G. Larson, and N. B. Kraje, "PC-Based Power Plant Performance Analysis," Proc. of EPRI Heat Rate Improvement Conference, GS-6635, p 6B-1, January 1990.

[17] Z. Guo, and R. E. Uhrig, "Use of Artificial Neural Networks to Analyze Nuclear Power Plant Performance," Nuclear Technology, vol. 99, p 36, July 1992.

[18] "Thermal Performance Diagnostics Manual for Nuclear Power Plants," Volumes 1,2,3, EPRI Report NP-4990, April 1987.

[19] D. M. Sopocy, R. E. Henry, S. M. Gehl, and S. M. Divakaruni, "Development of an On-Line Expert System : Heat Rate Degradation Expert System Advisor," Proc. Expert Systems Applications for the Electric Power Industry, Vol 2, p 911, Orlando, Florida, June 1989.

[20] M. McClintock, R. Metzinger, and N. Hirota, "Thermal Performance Advisor Expert System Development," Proc. Expert Systems Applications for the Electric Power Industry, Vol 2, p 1193, Orlando, Florida, June 1989.

[21] R. E. Uhrig, "Applications of Neural Networks to the Operation of Nuclear Power Plants." Proceedings of SMORN VI, Gatlinburg Tennessee, vol. 2, p 55.01, May 1991.

[22] E. Eryurek, B. R. Upadhyaya, and K. Kavaklioglu, "Hybrid Digital Signal Processing and Neural Network Applications in PWRs." Proceedings of AI91, Jackson, Wyoming, p 455, September 1991.

[23] B. R. Upadhyaya and E. Eryurek, "Application of Neural Networks for Sensor Validation and Plant Monitoring," Nuclear Technology, vol. 97, p 170, February 1992.

[24] E. Eryurek and B. R. Upadhyaya, "Sensor Validation for Nuclear Power Plants Using Adaptive Backpropagation Neural Network." IEEE Transactions on Nuclear Sciences, vol. 37, p 1040, April 1990.

[25] R. V. Miller, "Flow Measurement Engineering Handbook," p 6.20, McGraw-Hill, New York, 1989.

[26] D. Hasson, "Precipitation Fouling," Fouling of Heat Transfer Equipment, p 527, Hemisphere, New York, NY, 1981.

[27] A. Schlag, "Survey of Studies of Classical Venturi Meters at the University of Liege," Proceedings of Flow Measurement in Closed Conduits Symposium, vol. 1, p 271, Edinburgh, September 1960.

[28] F. V. A. Engel, "How to Improve Flow Measurement Accuracy of Orifices, Nozzles, and Venturi Meters," Proc of Flow Measurement in Closed Conduits Symposium, vol. 1, p 319, Edinburgh, September 1960.

[29] R. P. Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, vol. 4, p 4, April 1987.

[30] D. R. Hush and B. G. Horne, "Progress in Supervised Neural Networks," IEEE Signal Processing Magazine, p 8, January 1993.

[31] S. Y. Kung and J. N. Hwang, "An Algebraic Projection Analysis for Optimal Hidden Layer Units Size and Learning Rates in Backpropagation Learning," Proc IEEE Joint Conference on Neural Networks, p I-363, San Diego, California, July 1988.

[32] S. Y. Kung and Y. H. Hu, "A Frobenous Approximation Reduction Method (FARM) for Determining the Number of Hidden Units," Proc IEEE Joint Conference on Neural Networks, p II-163, Seattle, Washington, July 1991.

[33] J. Y. Choi and C. Choi, "Sensitivity Analysis of Multi-layer Perceptron with Differentiable Activation Functions." IEEE Transactions on Neural Networks, vol. 3, no. 1, p 101, January 1992.

[34] M. Iwatsuki, M. Kawamata and T. Higuchi, "Statistical Sensitivity and Minimum Sensitivity Structures with Fewer Coefficients in Discrete Time Linear Systems." IEEE Transactions on Circuits and Systems, vol. 37, no. 1, p 72, January 1988.

[35] "NeuralWorks Professional II/Plus and NeuralWorks Explorer," NeuralWare Inc. Pittsburgh, Pennsylvania, 1991.

APPENDICES

APPENDIX A

Code Listing for Steam Property Routines

```
/*
The following routine calculates the saturation temperature of water given the pressure
*/
float tsat(p)
 float p;
{
 float dummy;
 p=6.8946e-3*p;
 if (p < 0.002) {
   dummy=270.121*pow(.002,0.135019) - 99.2;
   return(1.8*dummy+32.0);
 }
 else if (p > 21.5) {
   dummy= 227.2963*pow(21.5,0.201581) - 50.0;
   return(1.8*dummy+32.0);
 }
 else if (p >= 0.002  &&  p < 0.01672) {
   dummy=270.121 * pow(p,0.135019) - 99.2;
   return(1.8*dummy+32.0);
 }
 else if (p >= 0.01672  &&  p < 0.0725) {
   dummy=254.6831 * pow(p,0.156108) - 78.2;
   return(1.8*dummy+32.0);
 }
 else if (p >= 0.0725 && p <= 0.359) {
   dummy=236.2315* pow(p,0.1784767) -57.0;
   return(1.8*dummy+32.0);
 }
 else if (p > 0.359 && p <= 1.676) {
   dummy=207.9248* pow(p,0.2092705) -28.0;
   return(1.8*dummy+32.0);
 }
 else if (p > 1.676 && p <= 8.511) {
```

```
    dummy=185.0779* pow(p,0.2323217) -5.0;
    return(1.8*dummy+32.0);
  }
  else if (p > 8.511 && p <= 17.69) {
    dummy=195.1819* pow(p,0.2241729) - 16.0;
    return(1.8*dummy+32.0);
  }
  else {
    dummy= 227.2963*pow(p,0.201581) - 50.0;
    return(1.8*dummy+32.0);
  }
}


/*
 The following routine calculates the saturation pressure of water given the pressure
*/
float psat(t)
 float t;
{
 float dummy;
 t=(t-32.0)/1.8;
 if (t  <  18.0) {
    dummy=pow(((18.0 + 99.2)/270.121),7.406365);
    return(145.0*dummy);
  }
 else if (t  >  373.0) {
    dummy=pow(((373.0+50.0)/227.2963),4.960785);
    return(145.0*dummy);
  }
 else if (t >= 18.0  &&  t < 56.275) {
    dummy=pow(((t+99.2)/270.121),7.406365);
    return(145.0*dummy);
  }
 else if (t >= 56.275  &&  t < 90.88) {
    dummy=pow(((t+78.2)/254.6831),6.4058216);
    return(145.0*dummy);
  }
 else if (t >= 90.88 && t < 139.781) {
    dummy=pow(((t+57.0)/236.2315),5.602972);
    return(145.0*dummy);
  }
 else if (t >= 139.781 && t < 203.662) {
```

```
    dummy=pow(((t+28.0)/207.9248),4.778504);
    return(145.0*dummy);
  }
  else if (t >= 203.662 && t <= 299.407) {
    dummy=pow(((t+5.0)/185.0779),4.304376);
    return(145.0*dummy);
  }
  else if (t > 299.407 && t < 355.636) {
    dummy=pow(((t+16.0)/195.1819),4.460843);
    return(145.0*dummy);
  }
  else {
    dummy=pow(((t+50.0)/227.2963),4.960785);
    return(145.0*dummy);
  }
}


/*
 The following routine calculates the enthalpy of the liquid phase of water in the saturated
region
*/
float hf(p)
  float p;
{
  float dummy;
  p=6.8946e-3*p;
  if (p < 0.002) {
    dummy= 1128.777 * pow(0.002,0.135196) - 413.72;
    return(0.42992*dummy);
  }
  else if (p  >  21.5) {
    dummy= 9.1417257e-14* pow(21.5,11.47287)+1752.0;
    return(0.42992*dummy);
  }
  else if (p >= 0.002  &&  p < 0.0173) {
    dummy= 1128.777 * pow(p,0.135196) - 413.72;
    return(0.42992*dummy);
  }
  else if (p >= 0.0173  &&  p < 0.1028) {
    dummy= 1050.7085 * pow(p,0.161797) - 306.5;
    return(0.42992*dummy);
  }
```

```
 else if (p >= 0.1028 && p < 0.942) {
   dummy= 912.1779* pow(p,0.2061637) -150.0;
   return(0.42992*dummy);
 }
 else if (p >= 0.942 && p < 4.02) {
   dummy= 638.0621* pow(p,0.2963192) +125.0;
   return(0.42992*dummy);
 }
 else if (p >= 4.02 && p < 9.964) {
   dummy= 373.7665* pow(p,0.4235532) +415.0;
   return(0.42992*dummy);
 }
 else if (p >= 9.964 && p < 16.673) {
   dummy= 75.38673* pow(p,0.8282384) +900.0;
   return(0.42992*dummy);
 }
 else if (p >= 16.673 && p < 20.396) {
   dummy= 0.1150827* pow(p,2.711412) +1440.0;
   return(0.42992*dummy);
 }
 else {
   dummy= 9.1417257e-14* pow(p,11.47287) +1752.0;
   return(0.42992*dummy);
 }
}


/*
 The following routine calculates the enhalpy of the vapor phase of water in the saturated
region
*/
float hg(p)
 float p;
{
 float dummy;
 p=6.8946e-3*p;
 if (p < 0.002) {
   dummy= 529.44008 * pow(0.002,0.108652) + 2263.5;
   return(0.42992*dummy);
 }
 else if (p > 21.5) {
   dummy= -22.92521*pow((21.55-18.0),2.0)+44.23671*(21.5-18.0) +2415.01;
   return(0.42992*dummy);
```

```
}
else if (p >= 0.002  &&  p < 0.1379) {
  dummy= 529.44008 * pow(p,0.108652) + 2263.5;
  return(0.42992*dummy);
}
else if (p >= 0.1379 && p <= 0.348) {
  dummy= -4.0381938e-6* pow((3.0-p),15.72364) +2750.0;
  return(0.42992*dummy);
}
else if (p > 0.348 && p <= 1.248) {
  dummy= -0.5767304* exp(-1.66153* (p-3.2)) +2800.0;
  return(0.42992*dummy);
}
else if (p > 1.248 && p < 2.955) {
  dummy= -7.835986* pow((3.001-p),2.) +2.934312*(3.001-p) +2803.71;
  return(0.42992*dummy);
}
else if (p >= 2.955 && p <= 6.522) {
  dummy=-1.347244*(p-2.999)*(p-2.999)-2.326913*(p-2.999)+2803.35;
  return(0.42992*dummy);
}
else if (p > 6.522 && p < 16.497) {
  dummy= -0.9219176* (p-9.0)*(p-9.0) -16.38835*(p-9.0) +2742.03;
  return(0.42992*dummy);
}
else if (p >= 16.497 && p < 20.193) {
  dummy= -3.532177* pow((p-8.0),2.0) +29.81305* (p-8.0) +2565.00;
  return(0.42992*dummy);
}
else {
  dummy= -22.92521* pow((p-18.0),2.0) +44.23671* (p-18.0) +2415.01;
  return(0.42992*dummy);
}
}
```

/*
The following routine calculates the specific volume of the liquid phase of water in the saturated region
*/

```
float vf(p)
```

```
 float p;
{
 float dummy;
 p=6.8946e-3*p;
 if (p < 0.002) {
   dummy=1.9118e-4 * pow(0.002,0.546472) + 0.0009947;
   return(dummy/0.06243);
 }
 else if (p > 21.5) {
   dummy= 4.1035988e-24* pow(21.5,15.03329) +0.00189;
   return(dummy/0.06243);
 }
 else if (p >= 0.002  &&  p < 0.01468) {
       dummy=1.9118e-4 * pow(p,0.546472) + 0.0009947;
   return(dummy/0.06243);
 }
 else if (p >= 0.01468  &&  p < 0.275) {
       dummy=1.380934e-4 * pow(p,0.388715) + 0.000987;
   return(dummy/0.06243);
 }
 else if (p >= 0.275 && p <= 1.0) {
   dummy=1.2746977e-4* pow(p,0.4644339) +0.001;
   return(dummy/0.06243);
 }
 else if (p > 1.0 && p <= 3.88) {
   dummy=1.0476071e-4* pow(p,0.5651090) +0.001022;
   return(dummy/0.06243);
 }
 else if (p > 3.88 && p <= 8.84) {
   dummy=3.2836717e-5* p +1.12174735e-3;
   return(dummy/0.06243);
 }
 else if (p > 8.84 && p <= 14.463) {
   dummy=3.3551046e-4 * exp(0.058403566 *p) +0.00085;
   return(dummy/0.06243);
 }
 else if (p > 14.463 && p < 18.052) {
   dummy= 3.1014626e-8* pow(p,3.284754) +0.00143;
   return(dummy/0.06243);
 }
 else if (p >= 18.052 && p < 20.204) {
   dummy= 1.5490787e-11* pow(p,5.7205) +0.001605;
   return(dummy/0.06243);
```

```
 }
 else {
   dummy= 4.1035988E-24* pow(p,15.03329) +0.00189;
   return(dummy/0.06243);
 }
}


/*
 The following routine calculates the enthalpy of water in the compressed liquid region
*/
float hfsub(p,t)
 float p,t;
{
 float pp,ps,tt,dummy;
 p=6.8946e-3*p;
 t=(t-32.0)/1.8;
 pp=p;
 tt=t;
 ps=psat(t*1.8+32.0)*6.8946e-3;
 if (p < ps) {
  pp=ps;
 }
 else if (p > 18.) {
  pp=18.;
 }
 else if (t < 91.79) {
  tt=91.79;
 }
 else if (t > 357.03) {
  tt=357.03;
 }
    dummy = hf(ps/6.8946e-3)/0.42992 + (1.4-169./(369.-tt))*(pp-ps);
    return(0.42992*dummy);
}
```

APPENDIX B

Code Listing for Heat Rate Calculation

```
/*****************************************************************

        This program evaluates the heat rate of a four-loop pressurized water reactor

*****************************************************************/

#include <stdio.h>
#include <math.h>
#include <c:\temp\prop.c>

/*****************************************************************

        The following functions are used as emprical data in the heat rate calculations
        poly1 : gives the percent reactor power as a function of feedwater flow
        poly2 : gives the expected condenser backpressure as a function of
                generator output and condenser inlet water temperature
        poly3 : gives the low-pressure turbine correction to load
                as a function of condenser steam flow and condenser backpressure
        poly4 : gives the low-pressure turbine correction to heat rate
                as a function of condenser steam flow and condenser backpressure
        poly5 : gives the steam pressure as a function of percent power
        poly6 : gives the pressure drop from steam generator to pressure taps
                as a function of feedwater flow

*****************************************************************/

float poly1(x)
 float x;
{
 float poly11;
 poly11 = 4.0364e-5*x*x*x - 6.31e-3*x*x + 1.46*x;
```

```
 return(poly11);
}

float poly2(x,y)
 float x,y;
{
 float poly22;
 poly22 = (((0.0605*(x-38.0)+1.12)-(0.0531*(x-38.0)+0.68))/(400.0))*(y-800.0)
       + (0.0531*(x-38.0) + 0.68);
 return(poly22);
}

float poly3(x,y)
 float x,y;
{
 float dum1,dum2,poly33;
 if (y <= 4.0){
   poly33 = 0.45*x*x - 6.888*x + 11.94143;
 }
 else if (y <= 6.0){
   dum1 = 0.45*x*x - 6.888*x + 11.94143;
   dum2 = -0.4357*x*x - 0.862*x + 3.33;
   poly33 = ((dum2-dum1)/(6.0-4.0))*(y-4.0) + dum1;
 }
 else if (y <= 8.0){
   dum1 = -0.4357*x*x - 0.862*x + 3.33;
   dum2 = -0.75*x*x + 1.798*x -0.80143;
   poly33 = ((dum2-dum1)/(8.0-6.0))*(y-6.0) + dum1;
 }
 else {poly33 = -0.75*x*x + 1.798*x -0.80143;}
 return(-poly33);
}

float poly4(x,y)
 float x,y;
{
 float poly44;
 poly44 = (-1)*poly3(x,y);
 return(poly44);
}

float poly5(x)
 float x;
```

```
{
 float poly55;
 poly55 = -2.086e-4*x*x*x + 4.3556e-2*x*x - 3.939*x + 997.7778;
 return(poly55);
}

float poly6(x)
 float x;
{
 float poly66;
 poly66 = 1.394e-2*x*x*x +0.386*x*x - 2.662e-3*x - 0.747576;
 return(poly66);
}


main() {

/************************************************************

                Define the Variables

************************************************************/

int number, unit, i;
float p2273a[100], u0442[100], u0474[100], u0482[100], p0496a[100],
 u0422[100], q0340a[100], p0398a[100], p0399a[100], u0484[100], u0485[100],
 t0752a[100], t0753a[100], t0754a[100], t0755a[100], t0702a[100], t0705a[100],
 t0708a[100], t0711a[100],
 p2277a[100], p2278a[100], p2279a[100], p2280a[100], p2281a[100], p2282a[100],
 t2320a[100], t2321a[100], t2322a[100], t2323a[100], t2324a[100], t2325a[100],
 u1118[100], u1169[100],
 t2430a[100], t2431a[100], t2434a[100], t2435a[100], t2438a[100], t2439a[100],
 repole[100], idgeno[100], avccwt[100], mavcbp[100], indcbp[100], cmacbp[100],
 expcbp[100], ecoclf[100], mcoclf[100], cochrf[100], mweecp[100], mwemcp[100],
 stthpr[100], destrp[100], mwetrp[100], msra1p[100], msrb1p[100], msrc1p[100],
 msra2p[100], msrb2p[100], msrc2p[100], msra1t[100], msrb1t[100], msrc1t[100],
 msra2t[100], msrb2t[100], msrc2t[100], tsata1[100], tsatb1[100], tsatc1[100],
 tsata2[100], tsatb2[100], tsatc2[100], ttda1[100], ttdb1[100], ttdc1[100],
 ttda2[100], ttdb2[100], ttdc2[100], avgttd[100], mwemsr[100], wsl[100],
 wht[100], wp[100], wcd[100], wba[100], wab[100], mwewsl[100],
 mwewht[100], mwewp[100], mwewcd[100], mwewba[100],
 mwewab[100], entwsl[100],
 entwht[100], entwp[100], entwcd[100], entwba[100], entwab[100], tprwsl[100],
```

```
 tprwht[100], tprwp[100], tprwcd[100], tprwba[100], tprwab[100], mweast[100],
 cofal1[2], cofal2[2], cofal3[2], cofal4[2], tcold[100], thot[100],
 wrwro[100], delhro[100],
 rcspr[100], entcl[100], enthl[100], qprim1[100], qprim2[100], qratio[100],
 vbdf[100], fwpra[100], sgpra[100], fwfr[100], entfwf[100], sgpcor[100],
 corsgp[100], entbdf[100], spvbdf[100], stfr[100], entlst[100], entvst[100],
 thrcf[100], ghrws[100], ghrvbd[100], ghrfwf[100], ghrrcp[100], ghr1[100],
 ghr2[100], cghr[100], wabno[100], dummy[100];

/*************************************************************************

                Input Output File Definitions

*************************************************************************/

FILE *infile, *outfile;
infile  = fopen("inp.txt","r");
outfile = fopen("out.txt","w");

/*************************************************************************

                Read the Input & Write to the Output

*************************************************************************/

fscanf(infile,"%d",&unit);
fprintf(outfile,"%d",unit);
fscanf(infile,"%d",&number);
fprintf(outfile,"%d",number);
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wsl[i]);
 fprintf(outfile,"%f ",wsl[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wht[i]);
 fprintf(outfile,"%f ",wht[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wp[i]);
 fprintf(outfile,"%f ",wp[i]);
```

```
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wcd[i]);
 fprintf(outfile,"%f ",wcd[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wba[i]);
 fprintf(outfile,"%f ",wba[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wab[i]);
 fprintf(outfile,"%f ",wab[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&wabno[i]);
 fprintf(outfile,"%f ",wabno[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&p2273a[i]);
 fprintf(outfile,"%f ",p2273a[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&u0442[i]);
 fprintf(outfile,"%f ",u0442[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&u0474[i]);
 fprintf(outfile,"%f ",u0474[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fscanf(infile,"%f",&u0482[i]);
 fprintf(outfile,"%f ",u0482[i]);
}
fprintf(outfile,"\n");
```

```
/*************************************************************************

                Reactor Power Level & Ideal Generator Output

*************************************************************************/

for(i=0; i<number; i++){
 repole[i] = 100.0*(u1118[i]/3423.0);
 if(repole[i] > 70.0)
   idgeno[i] = repole[i]*11.863;          /* in MWe */
 else{
   idgeno[i] = poly1(repole[i]);   /* this is in % */
   idgeno[i] = idgeno[i]*11.863;          /* this in MWe  */
 }
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",repole[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",idgeno[i]);
}
fprintf(outfile,"\n");

/*************************************************************************

                Condenser Performance

*************************************************************************/

for(i=0; i<number; i++){
 avccwt[i] = (t2430a[i]+t2431a[i]+t2434a[i]
 +t2435a[i]+t2438a[i]+t2439a[i])/6.0;
/*  cmacbp[i] = (mavcbp[i]+indcbp[i])/2.0; */
 expcbp[i] = poly2(avccwt[i],q0340a);
 ecoclf[i] = poly3(expcbp[i],8.0);
 mweecp[i] = q0340a[i]-(q0340a[i]/(1.0+ecoclf[i]/100.0));
 mcoclf[i] = poly3(mavcbp[i],8.0);
 mwemcp[i] = q0340a[i]-(q0340a[i]/(1.0+(mcoclf[i]/100.0-ecoclf[i]/100.0)));
 cochrf[i] = poly4(expcbp[i],8.0);
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",avccwt[i]);
```

```
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",mweecp[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",cochrf[i]);
}
fprintf(outfile,"\n");


/*************************************************************************

            Throttle Pressure Correction

*************************************************************************/

for(i=0; i<number; i++){
 stthpr[i] = p0496a[i] + 14.3;
 destrp[i] = poly5(repole[i]);
 mwetrp[i] = q0340a[i]-(q0340a[i]/(0.00015*(stthpr[i]-destrp[i])+1.0));
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",stthpr[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",destrp[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",mwetrp[i]);
}
fprintf(outfile,"\n");



/*************************************************************************

            Moisture Separator Reheater Improvement

*************************************************************************/
```

```
for(i=0; i<number; i++){
 msra1p[i] = p2277a[i] + 14.7;
 msrb1p[i] = p2278a[i] + 14.7;
 msrc1p[i] = p2279a[i] + 14.7;
 msra2p[i] = p2280a[i] + 14.7;
 msrb2p[i] = p2281a[i] + 14.7;
 msrc2p[i] = p2282a[i] + 14.7;
 tsata1[i] = tsat(msra1p[i]);
 tsatb1[i] = tsat(msrb1p[i]);
 tsatc1[i] = tsat(msrc1p[i]);
 tsata2[i] = tsat(msra2p[i]);
 tsatb2[i] = tsat(msrb2p[i]);
 tsatc2[i] = tsat(msrc2p[i]);
 ttda1[i]  = tsata1[i] - t2320a[i];
 ttdb1[i]  = tsatb1[i] - t2321a[i];
 ttdc1[i]  = tsatc1[i] - t2322a[i];
 ttda2[i]  = tsata2[i] - t2323a[i];
 ttdb2[i]  = tsatb2[i] - t2324a[i];
 ttdc2[i]  = tsatc2[i] - t2325a[i];
 avgttd[i] = (ttda1[i]+ttdb1[i]+ttdc1[i]+ttda2[i]+ttdb2[i]+ttdc2[i])/6.0;
 mwemsr[i] = (19.5-avgttd[i])*0.2;
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",msra1p[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",msrb1p[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",msrc1p[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",avgttd[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",mwemsr[i]);
}
fprintf(outfile,"\n");
```

```
/*********************************************************************

                Average T-hot & T-cold for Primary System

*********************************************************************/

cofal1[1]=59.92; cofal1[2]=59.988; cofal2[1]=60.41; cofal2[2]=60.529;
cofal3[1]=60.50; cofal3[2]=59.507; cofal4[1]=60.43; cofal4[2]=60.770;
for(i=0; i<number; i++){
 t0702a[i] = (t0702a[i]-1.0)*1.5*cofal1[unit]*(1.0/4.0);
 t0705a[i] = (t0705a[i]-1.0)*1.5*cofal2[unit]*(1.0/4.0);
 t0708a[i] = (t0708a[i]-1.0)*1.5*cofal3[unit]*(1.0/4.0);
 t0711a[i] = (t0711a[i]-1.0)*1.5*cofal4[unit]*(1.0/4.0);
 u0485[i]  = (t0702a[i]+t0705a[i]+t0708a[i]+t0711a[i])/4.0;
 t0752a[i] = (t0752a[i]-1.0)*25.0 + 530.0;
 t0753a[i] = (t0753a[i]-1.0)*25.0 + 530.0;
 t0754a[i] = (t0754a[i]-1.0)*25.0 + 530.0;
 t0755a[i] = (t0755a[i]-1.0)*25.0 + 530.0;
 u0484[i]  = (t0752a[i]+t0753a[i]+t0754a[i]+t0755a[i])/4.0;
 tcold[i]  = u0484[i]-(u0485[i]/2.0);
 thot[i]   = tcold[i]+u0485[i];
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",u0485[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",u0484[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",tcold[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",thot[i]);
}
fprintf(outfile,"\n");

/*********************************************************************

                Feedwater Venturi Fouling Factor
```

```
**********************************************************************/
wrwro[1]=wrwro[2]=1.0; delhro[1]=delhro[2]=79.95;
for(i=0; i<number; i++){
 rcspr[i]  = u0482[i] + 14.3;
 entcl[i]  = hfsub(rcspr[i],tcold[i]);
 enthl[i]  = hfsub(rcspr[i],thot[i]);
 qprim1[i] = wrwro[i]*((enthl[i]-entcl[i])/delhro[i])*3423.0;
 qprim2[i] = u1118[i] + 12.0;
 qratio[i] = qprim1[i]/qprim2[i];
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",enthl[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",qprim1[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",qprim2[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",qratio[i]);
}
fprintf(outfile,"\n");

/**********************************************************************

                Feedwater Evaluation

**********************************************************************/

for(i=0; i<number; i++){
  fwpra[i]  = p2273a[i] + 14.3;
  sgpra[i]  = u0474[i] + 14.3;
  fwfr[i]   = u0442[i]*1000.0;
  entfwf[i] = hfsub(fwpra[i],u0422[i]);
  sgpcor[i] = poly6(fwfr[i]/4.e6);
  corsgp[i] = sgpra[i] + sgpcor[i];
  entbdf[i] = tsat(corsgp[i]);
  spvbdf[i] = vf(corsgp[i]);
```

```
 vbdf[i]   = vbdf[i]*(8.022/spvbdf[i]);
 stfr[i]   = fwfr[i] - vbdf[i];
 entlst[i] = hf(corsgp[i]);
 entvst[i] = hg(corsgp[i]);
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",fwfr[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",sgpcor[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",corsgp[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",stfr[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",entlst[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",entvst[i]);
}
fprintf(outfile,"\n");
```

```
/*************************************************************************

                Heat Rate Evaluation

*************************************************************************/

for(i=0; i<number; i++){
 dummy[i]  = q0340a[i]*1000.0;
 ghrws[i]  = (stfr[i]*(0.9975*entvst[i]+0.0025*entlst[i]))/(dummy[i]);
```

```
 ghrvbd[i] = (vbdf[i]*entbdf[i])/dummy[i];
 ghrfwf[i] = (fwfr[i]*entfwf[i])/dummy[i];
 ghrrcp[i] = (12000.0*3412.7)/dummy[i];
 ghr1[i]  = ghrws[i] + ghrvbd[i] - ghrfwf[i] - ghrrcp[i];
 ghr2[i]  = (u1118[i]/q0340a[i])*3412.7;
 cghr[i]  = (ghr2[i])/(1.0+(cochrf[i]/100.0));
}
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghrws[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghrvbd[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghrfwf[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghrrcp[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghr1[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",ghr2[i]);
}
fprintf(outfile,"\n");
for(i=0; i<number; i++){
 fprintf(outfile,"%f ",cghr[i]);
}
fprintf(outfile,"\n");
fclose(infile);
fclose(outfile);
}
```

APPENDIX C

Code Listing For Feedwater Heater Efficiency Calculation

```
/*************************************************************************

This program calculates the thermal efficiency of the feedwater heater

*************************************************************************/


#include <stdio.h>
#include <math.h>
#include "prop.c"

#define FSTEAM  34.71
#define FFEED  2866.39
#define FPRES  167.00

main ()
{
 int i;
 double qsteam, qfeed, perf;
 double hs1, hs2, hs3, hd, hfin, hfout;
 /*
 double P2239A, P2240A, P2241A, T2225A, T2226A, T2227A;
 double T2246A, T2247A, T2248A, T2269A, T2270A, T2271A;
 */
 double input[12];

 FILE *infile, *outfile, *testfile;

 infile  = fopen("inp.txt","r");
 outfile  = fopen("out.txt","w");
 testfile = fopen("tes.txt","w");

 for (;;)
```

```
{

 if (feof(infile) != 0) exit (0);


 for (i=0;i<12;i++)
 {
  fscanf(infile,"%lf", &input[i]);
  fprintf(testfile, "%lf\t", input[i]);
 }
 fprintf(testfile, "\n");

 input[3]=psat(input[3]);
 input[4]=psat(input[4]);
 input[5]=psat(input[5]);


 qsteam = FSTEAM*(hg(input[0])-hf(input[3])+hg(input[1])-hf(input[4])
            +hg(input[2])-hf(input[5]));

 qfeed  = (FFEED/3.0)*(hfsub(FPRES,input[9])-hfsub(FPRES,input[6])
                +hfsub(FPRES,input[10])-hfsub(FPRES,input[7])
                +hfsub(FPRES,input[11])-hfsub(FPRES,input[8]) );

 perf=qfeed/qsteam;

 fprintf(outfile, "%lf\t%lf\t%lf\n",qfeed,qsteam,perf);

 }

fclose(infile);
fclose(outfile);
fclose(testfile);

}
```

VITA

Kadir Kavaklioglu was born in Denizli, Turkiye on January 7, 1968. He attended Ressam Ibrahim Calli Primary School, Cumhuriyet Middle School, and Denizli High School in Denizli, and graduated from high school in June 1984. The following September, he entered the Nuclear Engineering Department of Hacettepe University in Ankara, Turkiye. He received his Bachelor of Science degree in Nuclear Engineering in June 1989. Between September 1989 and December 1990, he worked as a research assistant for the same department. In January 1991, he entered the Nuclear Engineering Department of The University of Tennessee, and in May 1993, he received his Master of Science degree in Nuclear Engineering. He is presently a research assistant and a Ph. D. candidate in the same department.