12-1991

# Application of neural networks to measurement of temperature sensor response time

Agus Cahyono

To the Graduate Council:

I am submitting herewith a thesis written by Agus Cahyono entitled "Application of neural networks to measurement of temperature sensor response time." I have examined the final electronic 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.

E. M. Katz, Major Professor

We have read this thesis and recommend its acceptance:

R. E. Uhrig, L. F. Miller

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

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

To the Graduate Council:

I am submitting herewith a thesis written by Agus Cahyono entitled "Application of Neural Networks To Measurement of Temperature Sensor Response Time." 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.

E. M. Katz, Major Professor

We have read this thesis
and recommend its acceptance:

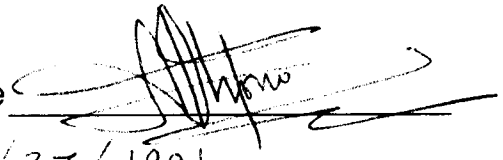
Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

# STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master of Science degree at The University of Tennessee, Knoxville, I agree that Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by major professor, or in her absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature 

Date 11/27/1991

# APPLICATION OF NEURAL NETWORKS TO MEASUREMENT OF TEMPERATURE SENSOR RESPONSE TIME

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Agus Cahyono

December 1991

# ACKNOWLEDGMENTS

The author would like to thank his major professor, DR. E. M. Katz, for her valuable guidance and suggestions throughout the development of this thesis and for her financial support in the form of research assistantship.

The author would also like to thank the other committee members, DR. R. E. Uhrig and DR. L. F. Miller, for their comments and assistance during the course of this research.

He would also like to thank DR. T. W. Kerlin, advisor and Head of The Department of Nuclear Engineering, for his helpful advice in the completion of this research.

Many thanks are given to Mr. R. Bailey and Mr. G. Graves for their assistance in the experiments of this research.

The help of Mr. L. Mullinix in performing the experiments is really appreciated.

# ABSTRACT

A neural network, implementing the backpropagation paradigm, has been developed to predict the time constants of resistance temperature detectors (RTDs) from loop current step response (LCSR) test transients. It eliminates the difficulties involved in the LCSR application: complicated computation, specialized equipment, and highly trained personnel.

The neural network consists of three fully connected layers: an input layer, a hidden layer, and an output layer, with the number of input-layer processing elements (PEs) is varied from 20 to 60. The best results are obtained by the network consisting of 60 input-layer PEs, 150 hidden-layer PEs, and 1 output-layer PE.

A series of LCSR tests on 2 RTDs, the type of sensor used in most pressurized water reactors (PWRs) to trip safety systems, generates the response transients of the sensors, the input data of the networks. Plunge tests are used to determined the time constants of the RTDs, the desired output of the neural networks. Neural networks have been trained using these sets of input/output data from one RTD. The trained networks are used to predict the time constant of the other RTD. The time constant predictions of the trained networks produce the average relative error of about 5 percent.

In order to identify the network's sensitivity, tests with imperfect equipment have been performed to generate imprecise

LCSR data and other tests use the LCSR data to which simulated noise has been added to the LCSR data.

The time constant predictions of the networks using the test sets of imprecise data produce the average relative error of about 6 percent. The average relative error of the time constant predictions of the networks using the test sets of data contaminated with 3 and 5 percent noise is within 8 percent. This indicates that backpropagation networks have been able to overcome contaminated data and equipment imperfections.

# TABLE OF CONTENTS


| Chapter | | Page |
|---|---|---|
| 1 | INTRODUCTION | 1 |
| | • Organization Of The Text | 2 |
| | • Literature Review | 3 |
| 2 | RESISTANCE THERMOMETRY | 5 |
| | • Introduction | 5 |
| | • RTD Description | 7 |
| | • RTD's Installation | 9 |
| | • Resistance Measurement | 13 |
| | • Sensor Response Time | 22 |
| 3 | CURRENT METHODS OF ESTIMATING THE TIME CONSTANTS OF RTDs | 27 |
| | • LCSR Test | 27 |
| | • Plunge Test | 34 |
| | • Sampling Data | 40 |
| 4 | NEURAL NETWORKS | 45 |
| | • Introduction | 45 |
| | • Backpropagation Neural Networks | 49 |
| 5 | OPTIMIZING BACKPROPAGATION NETWORKS | 60 |
| | • Introduction | 60 |
| | • Momentum Term | 60 |
| | • Selecting The Optimum Number Of Layers | 61 |

- Selecting The Optimum Number Of PEs 61

6 NEURAL NETWORK METHODS 63

- Introduction 63
- Neural Network Software Evaluation 63

7 RESULTS 68

- Training Neural Networks 68
- Recall Neural Networks 74
- Predictions Of Neural Networks Using Perturbed Sets Of Data 89
- Predictions Of Neural Networks Using Sets Of Data Obtained From Imperfect Equipment 93

8 CONCLUSIONS 99

LIST OF REFERENCES 101

APPENDIXES 106

- Appendix A. $R_S$ Computation 107
- Appendix B. Program For Selecting LCSR Data 110
- Appendix C. Program FOR Simulating Noise 111
- Appendix D. Program Of Sampling Interfaces 113

VITA 118

# LIST OF TABLES


| Table | | Page |
|---|---|---|
| 2.1 | Platinum Characteristics | 6 |
| 3.1 | LCSR Test Specifications | 35 |
| 3.2 | Plunge Test Specifications | 39 |
| 3.3 | The Results Of Plunge Tests On RTD A | 42 |
| 3.4 | The Results Of Plunge Tests On RTD B | 43 |
| 7.1 | Neural Network Specifications | 70 |

# LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 2.1 | A Wall-mount RTD | 8 |
| 2.2 | A Typical RTD Used In Nuclear Reactors | 10 |
| 2.3 | A Typical Installation Of A Well-type RTD | 11 |
| 2.4 | A Typical Installation Of A Wet-type RTD | 12 |
| 2.5 | A Typical Wheatstone Bridge | 14 |
| 2.6 | A Wheatstone Bridge With Lead Wires | 18 |
| 2.7 | A Wheatstone Bridge With A Three-wire RTD Connection | 20 |
| 2.8 | A Wheatstone Bridge With A Four-wire RTD Connection | 21 |
| 2.9 | Circuit For A Four-wire RTD | 23 |
| 2.10 | Time Constant Measurement From Plunge Test Transient | 26 |
| 3.1 | A Typical LCSR Transient | 29 |
| 3.2 | An Equipment Schematic For An LCSR Test | 33 |
| 3.3 | An Equipment Schematic For A Plunge Test | 37 |
| 3.4 | Steps To Obtain Text Data File | 44 |
| 4.1 | A Processing Element's Operation | 46 |
| 4.2 | An Example Of A Three-layer Neural Network Architecture | 48 |
| 4.3 | A Typical Three-layer Backpropagation Network For Time Constant Prediction | 50 |
| 4.4 | Weight Changes In The Generalized Delta Rule | 51 |
| 4.5 | A Sigmoid Function Curve | 52 |
| 4.6 | A Curve Of A Sigmoid Function Derivative | 54 |

4.7 Two Step Procedures Of Training In A Backpropagation Network 55
4.8 Schematic For Modifying An Output-layer Weight 57
4.9 Schematic For Modifying A Hidden-layer Weight 59
6.1 Comparison Study Between NeuralWorks And ANSim 67
7.1 Neural Network Training Procedure For Measurement Of Temperature Sensor Response Time 69
7.2 An Example Of 20 Input Data For Training Neural Networks 71
7.3 An Example Of 40 Input Data For Training Neural Networks 72
7.4 An Example Of 60 Input Data For Training Neural Networks 73
7.5 The Average Relative Error Of The Time Constant Predictions Of The Networks Using Training Sets Of Data 75
7.6 Time Constant Predictions Of Network A Using Test Sets Of Data 77
7.7 Time Constant Predictions Of Network B Using Test Sets Of Data 78
7.8 Time Constant Predictions Of Network C Using Test Sets Of Data 79
7.9 Time Constant Predictions Of Network D Using Test Sets Of Data 80
7.10 Time Constant Predictions Of Network E Using Test Sets Of Data 81

7.11 Time Constant Predictions Of Network F Using Test Sets Of Data 82
7.12 Time Constant Predictions Of Network G Using Test Sets Of Data 83
7.13 Time Constant Predictions Of Network H Using Test Sets Of Data 84
7.14 Time Constant Predictions Of Network I Using Test Sets Of Data 85
7.15 Time Constant Predictions Of Network J Using Test Sets Of Data 86
7.16 Time Constant Predictions Of Network K Using Test Sets Of Data 87
7.17 The Average Relative Error Of The Time Constant Predictions Of The Networks Using Test Sets Of Data 88
7.18 An Example Of 60 Input Data Contaminated With Noise For Training The Network 90
7.19 Time Constant Predictions Of The Network Using Training Sets Of Data Contaminated With 3 Percent Noise 91
7.20 Time Constant Predictions Of The Network Using Training Sets Of Data Contaminated With 5 Percent Noise 92
7.21 Time Constant Predictions Of The Network Using Test Sets Of Data Contaminated With 3 Percent Noise 94

7.22 Time Constant Predictions Of The Network Using Test Sets Of Data Contaminated With 5 Percent Noise 95

7.23 Time Constant Predictions Of The Network Using The Training Sets Of Input/Output Data Obtained From Imperfect Equipment 96

7.24 Time Constant Predictions Of The Network Using Test Sets Of Input/Output Data Obtained From Imperfect Equipment 97

A.1 Wheatstone Bridge Equipped With A Switch And Resistor $R_S$ 108

# CHAPTER 1
# INTRODUCTION

One of the important components in nuclear reactor safety systems is temperature measurement. A temperature transient must be detected rapidly so that timely control and safety actions can be taken. Because of the importance of the speed of the sensor's response, it is necessary to verify it for each temperature sensor throughout the life of a nuclear power plant. In pressurized water reactors (PWRs), resistance temperature detectors (RTDs) are used to monitor temperature.

The response characteristics of an RTD are represented commonly by the time constant. It is defined as time required to achieve 63.2 percent of the change in temperature following a step change in the input.

The time constant of an RTD can be measured in a laboratory by a plunge test. This method is based on an external step change in temperature caused by a sudden immersion of the warmed RTD from air into cold water. The time constant of the RTD is determined from the response transient. The results of the plunge test performed in a laboratory may not reflect the time constant of an RTD installed in nuclear power plant, because the time constant of an RTD is affected by its environment. An in-situ testing method called the loop current step response (LCSR) test [1] has been developed to measure the response times of RTDs installed in a nuclear power plant.

However, application of an LCSR test involves complicated computation, highly trained personnel, and specialized equipment to interpret the LCSR data and to obtain the time constant of the RTD. Because of these difficulties, it was proposed to utilize a back-propagation neural network to predict the time constant from LCSR response transients.

This research has developed such a method, which eliminates the difficulties involved in the LCSR application and increases the accuracy of the estimated time constants by employing neural networks to predict the time constants of the RTDs.

## ❒ ORGANIZATION OF THE TEXT

Chapter 2 discusses resistance temperature detectors (RTDs), resistance measurement, and the time constant of RTDs and its measurement. Chapter 3 presents current methods of estimating the time constants of RTDs: LCSR test and plunge test. Chapter 4 introduces the theory of neural networks and the operation of backpropagation networks. Chapter 5 describes the methods to optimize the backpropagation network. Chapter 6 presents neural network methods to predict the time constant of the RTD. Chapter 7 discusses the results of the neural network methods in predicting the time constant. Chapter 8 presents the conclusions of the neural network application to measurement of temperature sensor reponse time.

## ❒ LITERATURE REVIEW

Because of the importance of temperature sensor response time in nuclear reactor safety systems, the temperature sensor response characteristics have been investigated over the last two decades. A method called the loop current step response (LCSR) test for an in-situ measurement of thermocouple response time was proposed in 1975 [1].

In the LCSR test, heat is generated at the sensing wire, and then it diffuses through the sensor assembly to the surrounding fluid. In the plunge test, heat must diffuse through the sensor assembly to the sensing wire. Since the response transient of the sensor to an external perturbation is desired and the response transient of the installed sensor to an internal perturbation can be obtained by the LCSR test (described in Chapter 3), a transformation is needed to convert the internal heating transient into external heating transient. This transformation considers the fact that heat transfer resistances and heat capacities of the sensor assembly are independent of the direction of heat flow. An analytical transformation required to convert LCSR test results into external heating transients from which the sensor time constant can be measured has been developed by Kerlin, et al. [2].

Hashemian [3] applied the LCSR test to RTDs installed in a nuclear power plant to evaluate quantitatively the response time of those sensors. He also analyzed the sensor response time degradation by implementing the self-heating test, which is based on the steady state measurement of the temperature rise in a sensor as a

function of electric power input. The maximum error of the time constant estimated from the transformation of the LCSR test data was about 20 percent.

A correlation, presented by Poore [4], resulted from further analytical and numerical studies has reduced that error to within 10 percent. However, the correlation involves complicated computation to obtain an accurate solution.

# CHAPTER 2
# RESISTANCE THERMOMETRY

## ❒ INTRODUCTION

The resistance of pure metals increases with temperature. The nearly linear relationship between resistance and temperature has led to the application of a metal as a sensing element in a temperature measu-rement device. Other desirable properties of a metal used as sensing elements are

- high resistivity,
- high melting point,
- high tensile strength and ductility,
- physical and chemical stability.

Based on these properties, platinum is considered the most suitable metal for a sensing element. Table 2.1 lists some important properties of platinum [9]. The resistance-temperature relationship for platinum is expressed by the Callender-Van Dusen equation [7] :

$$R/R_0 = 1 + \alpha [ T - \delta(0.01T - 1)(0.01T) - \beta(0.01T - 1)(0.01T)^3 ] \quad (2.1)$$

where

T = the temperature ($^0$C),

R = the resistance at temperature T ($\Omega$),

$R_0$ = the resistance at ice point ($\Omega$),

$\alpha$ = a constant (gives the linear approximation to the R versus T curve),

Table 2.1. Platinum Characteristics

| Characteristics | Value |
|---|---|
| Temperature Coeff. of Resistance | 0.00392 Ω/Ω $^0$C from 0 to 1000 $^0$C |
| Resistivity | 60.0 Ω/cir mil* at 0 $^0$C |
| Useful temperature range | -258 $^0$C to 900 $^0$C |
| Minimum practical wire diameter | 0.002 in. |
| Tensile strength | 18,000 psi annealed |

* A circular mil is the area contained in a circle with diameter of 0.001 inches.

$\delta$ = a constant depends on the material,

$\beta$ = a constant (zero when T is greater than 0 $^0$C).

The platinum sensing elements are usually used in resistance temperature detectors (RTDs).

## ❒ RTD DESCRIPTION

An industrial RTD consists of a sensing element, a supporting structure, a protecting sheath, and lead wires. The RTD sensing element is made from platinum wires. The sensing wire is wound in a form of coil around a supporting structure to minimize strain on the wire. The supporting structure may be made from glass [7]. In some designs, the platinum wire coil is cemented to the inner surface of a protecting sheath. An RTD designed in this type is called a wall-mount RTD as shown in Figure 2.1. The cement serves to support the coil and to provide electrical insulation between the coil and the protecting sheath.

The sensing wire wound on a supporting structure is then mounted inside a metalic protecting sheath that is usually stainless steel. The protecting sheath is evacuated and filled with powder. An ideal powder would have the following properties [7]:

- high electrical resistivity,
- high thermal conductivity,
- low abrasiveness (so thermal cycling cannot cause cutting into the sensing element or lead wires),
- chemical inertness relative to other sensor components,
- low affinity for absorbing water vapor.

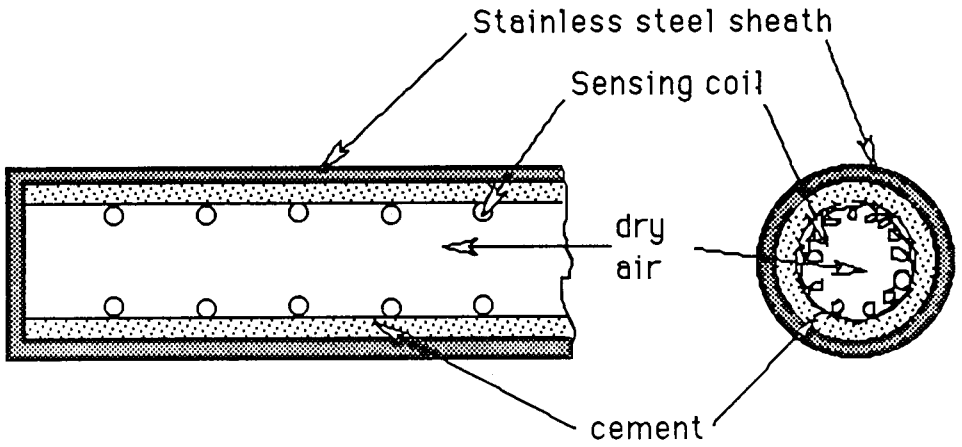



Figure 2.1. A Wall-mount RTD

In most applications, magnesium oxide or aluminum oxide powders are chosen to fill the protecting sheath. The powder is used to isolate any electrical contact between the sensing wire and the sheath. The powder also prevents any contanct between the lead wires, which are usually made from platinum. A Typical RTD used in nuclear reactors is shown in Figure 2.2.

## ❒ RTD's INSTALLATION

Based on their installation in nuclear reactors, RTDs can be classified in two types: well-type sensor and wet-type sensor. A well-type sensor is installed in a stainless steel thermowell secured to the coolant piping as shown in Figure 2.3. This installation protects the RTD from corrosive materials, pressure, and the force of the coolant flow. The well-type sensor can be inspected and replaced easily without interfering with the coolant operation. However, temperature response of the well-type sensor is slower, because the heat must pass through the thermowell also, affecting the coolant temperature measurement [3].

A wet-type sensor is installed directly in the flowing coolant without a thermowell as shown in Figure 2.4 [3]. The temperature response time measured by this wet-type sensor is faster than that measured by the well-type sensor. The disadvantages of the wet installation are that

- the sensor replacement is difficult,
- the sensor experiences the force and pressure of the coolant,
- the sensor might corrode.

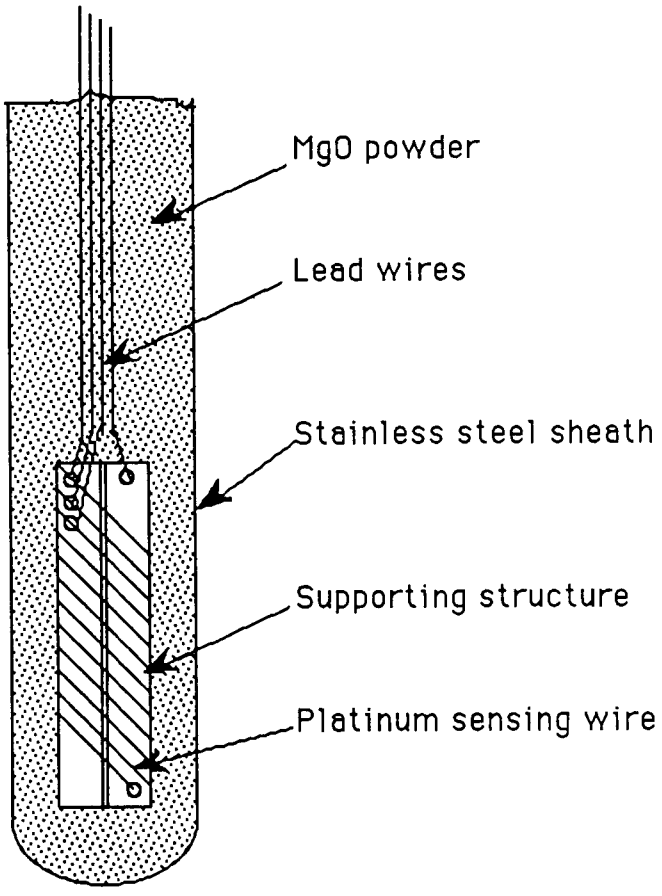



Figure 2.2. A Typical RTD Used In Nuclear Reactors

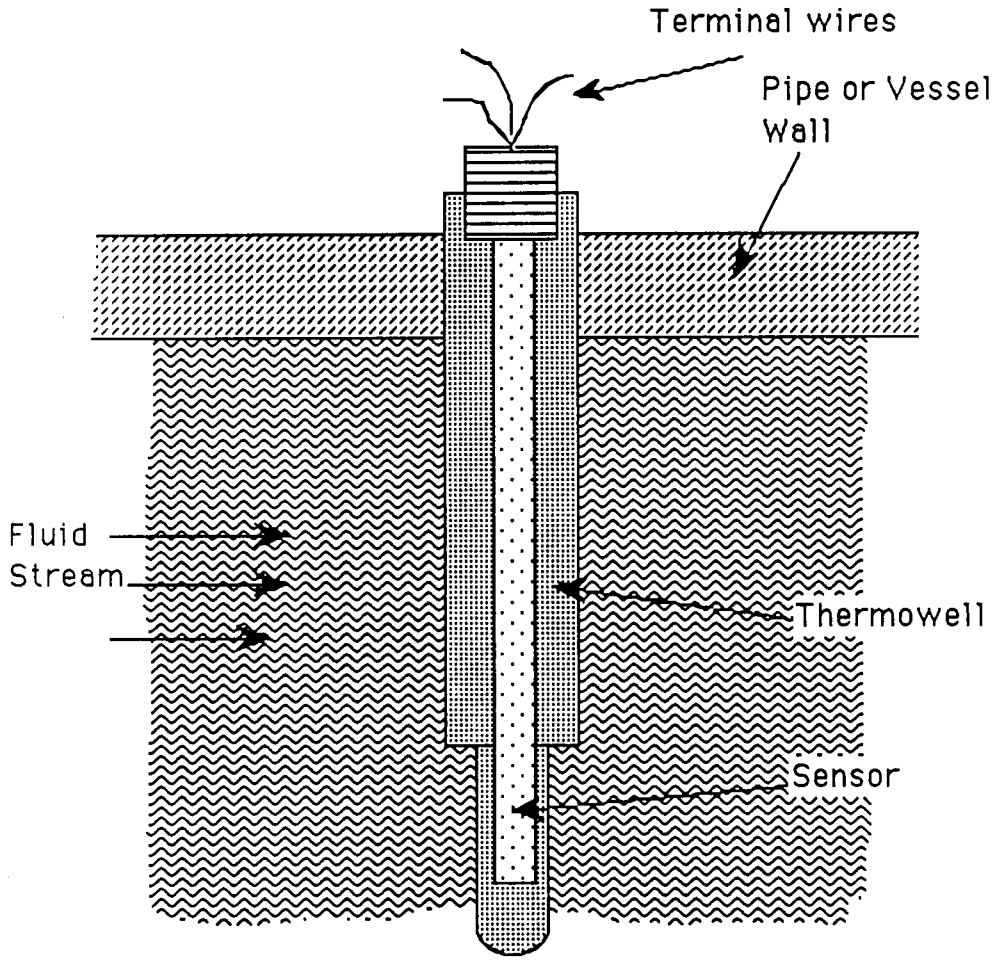



Figure 2.3. A Typical Installation Of A Well-type RTD [3]

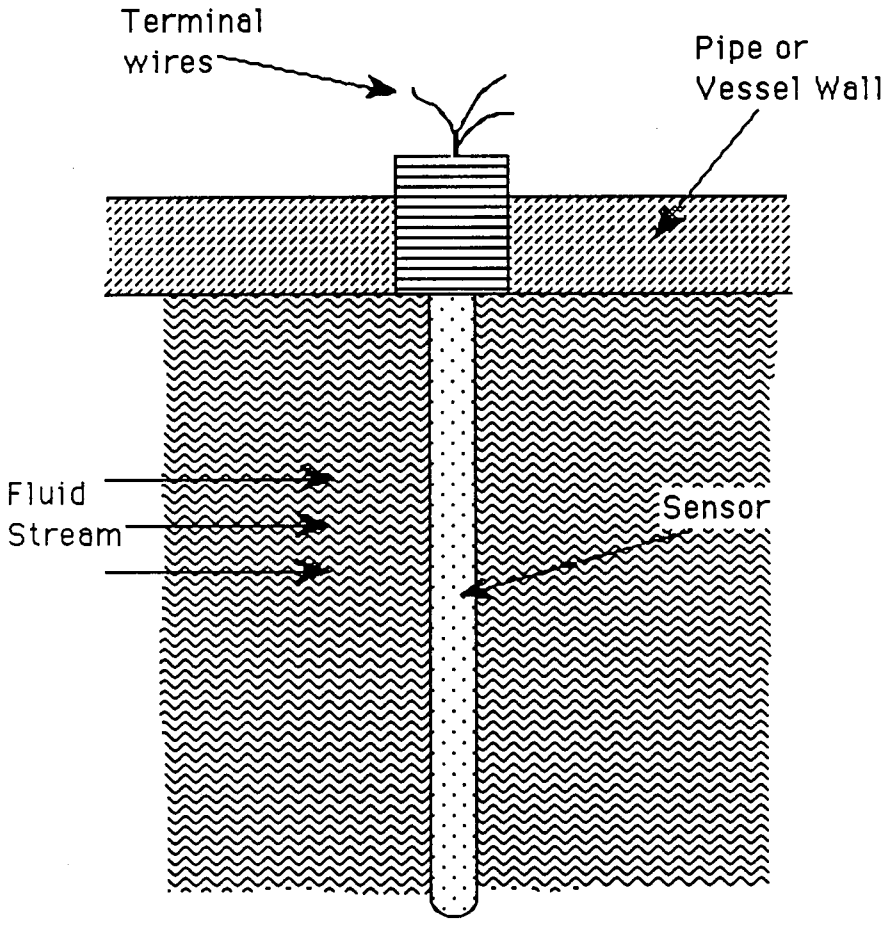



Figure 2.4. A Typical Installation Of A Wet Type RTD [3]

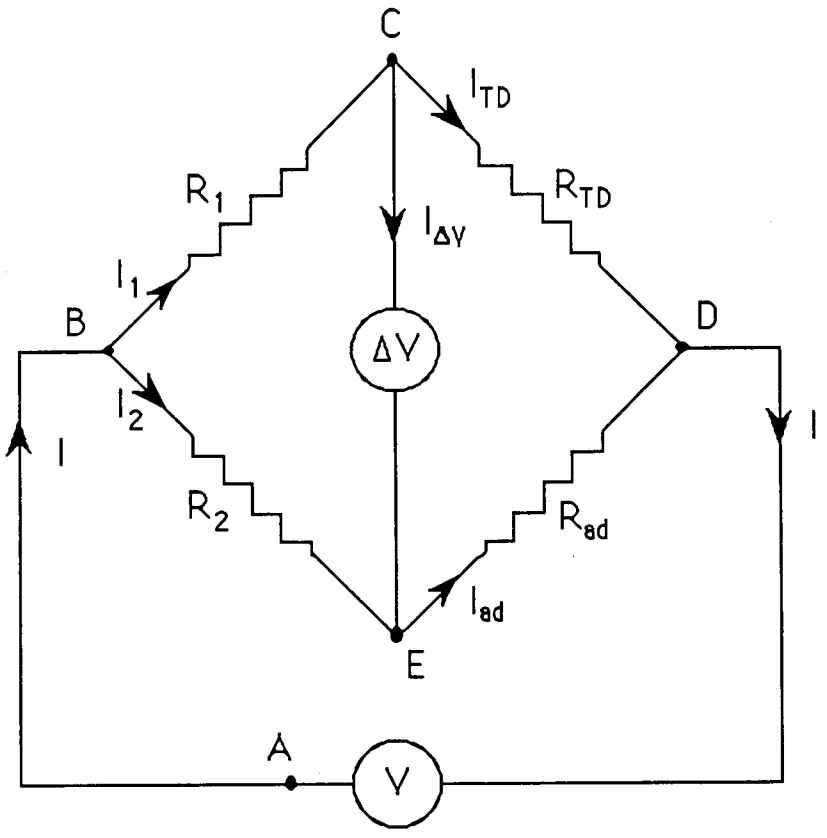
## ❒ RESISTANCE MEASUREMENT

In temperature measurements, resistance of a sensing element changes proportionally with temperature. This indicates that measuring the sensor resistance is required. The sensor resistance can be measured by using a Wheatstone bridge.

### ❖ Wheatstone Bridge

This Wheatstone Bridge was invented by Samuel Hunter Christie in 1833. Its application to electrical measurements was started by Sir Charles Wheatstone in 1843 [12]. A typical Wheatstone bridge, as shown in Figure 2.5, consists of four resistance elements: two fixed resistors, one adjustable resistor, and the RTD resistance to be measured. These resistors are connected to a power supply, V. The connection then yields an output voltage $\Delta V$.

The resistance of the RTD can be measured by either nonbalancing or balancing the bridge. In the nonbalancing method, the adjustable resistor $R_{ad}$ is fixed, and the bridge is not in balance because of changes in the resistance temperature detector $R_{TD}$. The imbalance produces changes in the output voltage $\Delta V$ that is ideally would be directly proportional to $R_{TD}$. The resistance $R_{TD}$ may be determined by measuring the output voltage $\Delta V$. However, this direct proportionality does not occur. It is approximated by the following equation [7]:




$R_1$ = resistor 1
$R_2$ = resistor 2
$R_{TD}$ = resistance temperature detector
$R_{ad}$ = adjustable resistor
V = voltage
ΔV = output voltage

$I_1$ = current through resistor 1
$I_2$ = current through resistor 2
$I_{TD}$ = current through RTD
$I_{ad}$ = current through adjustable resistor
$I_{\Delta V}$ = output current

Figure 2.5. A Typical Wheatstone Bridge

$$\Delta V = V\left[\frac{R}{(R + R_{TD})\,(R + R_{To})}\,(R_{TD} - R_{To})\right], \qquad (2.2)$$

where

$V$ = the power supply voltage,

$\Delta V$ = the output voltage,

$R = R_1 = R_2$ = a resistor,

$R_{TD}$ = the resistance temperature detector,

$R_{To}$ = the resistance of $R_{TD}$ at a reference temperature (i.e., $0^0$C).

Equation (2.2) shows that the relationship between E and $R_{TD}$ is nearly linear if R is large.

In the balancing method, the adjustable resistor is changed until the output $\Delta V$ goes to zero. In this case, the resistance of the RTD is equal to the resistance of the adjustable resistor. The balancing method is discussed below.

The relationship between power supply, V, and the output, $\Delta V$, can be obtained by applying Kirchoff's laws. The laws state that the algebraic sum of the voltage drops around a closed loop is zero and that sum of the currents flowing into a junction is equal to the sum of those flowing away from it. By using Figure 2.5 and applying Kirchoff's Laws, one can obtain

at point C:

$$I_{TD} = I_1 - I_{\Delta V}\,, \qquad (2.3)$$

at point E:

$$I_{ad} = I_2 + I_{\Delta V}\,, \qquad (2.4)$$

loop ABCDA:

$$0 = -V + I_1R_1 + I_{TD}R_{TD} \quad \text{or} \quad V = I_1R_1 + I_{TD}R_{TD}\,, \tag{2.5}$$

loop ABEDA:

$$0 = -V + I_2R_2 + I_{ad}R_{ad} \quad \text{or} \quad V = I_2R_2 + I_{ad}R_{ad} \tag{2.6}$$

loop CEDC:

$$0 = \Delta V + I_{ad}R_{ad} - I_{TD}R_{TD} \quad \text{or} \quad \Delta V = I_{TD}R_{TD} - I_{ad}R_{ad}\,, \tag{2.7}$$

where

$V$ = the power supply voltage,

$\Delta V$ = the output voltage,

$R_1$ = the resistor 1,

$R_2$ = the resistor 2,

$R_{ad}$ = the adjustable resistor,

$R_{TD}$ = the resistance temperature detector,

$I_{\Delta V}$ = the output current,

$I_1$ = the current through the resistor 1,

$I_2$ = the current through the resistor 2,

$I_{ad}$ = the current through the adjustable resistor,

$I_{TD}$ = the current through the RTD.

By combining Equations (2.3) through (2.7), hence

$$\Delta V = \frac{V\,(R_2R_{TD} - R_1R_{ad})}{(R_1 + R_{TD})\,(R_2 + R_{ad})} - I_{\Delta V}\left\{\frac{(R_1R_{TD})}{(R_1 + R_{TD})} - \frac{(R_2R_{ad})}{(R_2 + R_{ad})}\right\}. \tag{2.8}$$

If $I_{\Delta V} = 0$,

$$\Delta V = \frac{V\,(R_2R_{TD} - R_1R_{ad})}{(R_1 + R_{TD})\,(R_2 + R_{ad})}\,. \tag{2.9}$$

If the bridge is in balance, $\Delta V = 0$, and Equation (2.9) becomes

$$R_2R_{TD} = R_1R_{ad} \qquad \text{or} \qquad R_{TD} = R_1R_{ad}/R_2 . \tag{2.10}$$

The resistance of the RTD is obtained from Equation (2.10).

### ❖ The Lead Wire Effect

In a nuclear power plant, the RTD may be located relatively far from the measuring bridge. In order to measure the RTD resistance, the RTD is connected to the bridge by lead wires, which are usually copper. The resistance of the lead wires can cause an error in temperature measurement, since their resistance can be significant compared to the sensor resistance. Figure 2.6 shows an RTD with lead wires connected to the Wheatstone Bridge.

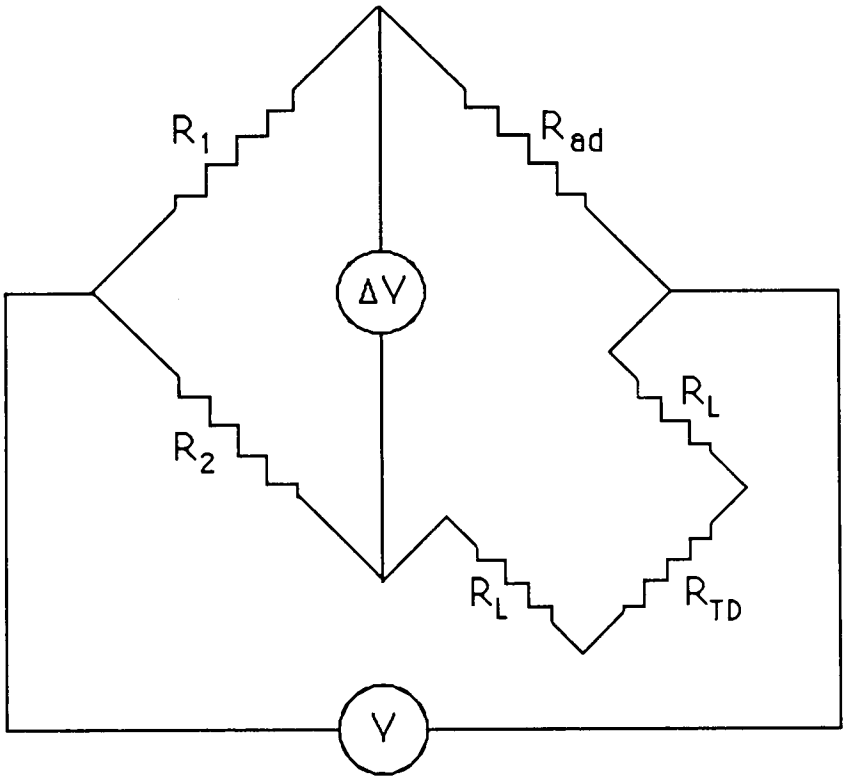
When the lead wires are used in temperature measurements, and $R = R_1 = R_2$, Equation (2.9) becomes

$$\Delta V = V \left\{ \frac{R\,(R_{TD} - R_{ad} + 2R_L)}{(R + R_{TD} + 2R_L)\,(R + R_{ad})} \right\} . \tag{2.11}$$

Equation (2.11) shows that the output $\Delta V$ includes the effect of the lead wire resistance, $R_L$.

### ❖ Three-Wire RTD

The three-wire RTD is widely used in industrial applications to avoid the effect of lead wires. The Wheatstone bridge configu-




$R_1$ = resistor 1
$R_2$ = resistor 2
$R_{TD}$ = resistance temperature detector
$R_{ad}$ = adjustable resistor
$R_L$ = lead wire resistance
V = voltage

Figure 2.6. A Wheatstone Bridge With Lead Wires

ration using the three-wire RTD is shown in Figure 2.7. The bridge output is

$$\Delta V = V\left\{\frac{(R_{TD} + R_L)}{(R + R_{TD} + R_L)} - \frac{(R_{ad} + R_L)}{(R + R_{ad} + R_L)}\right\}. \qquad (2.12)$$

When the bridge is in balance, $\Delta V = 0$, then

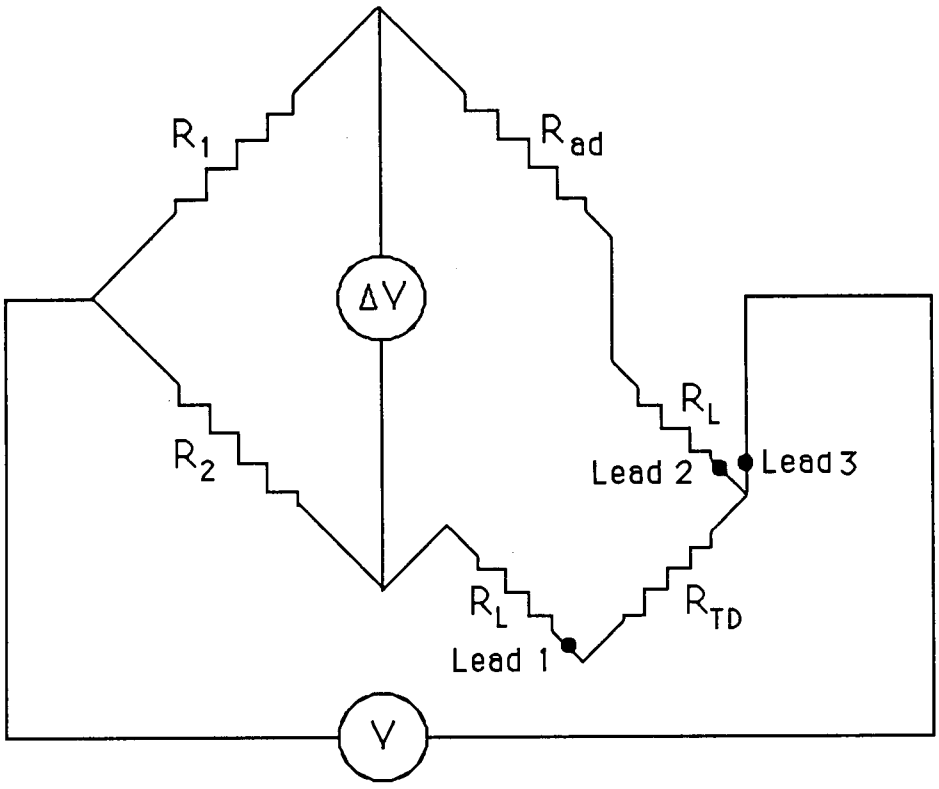
$$R_{TD} + R_L = R_{ad} + R_L \qquad \text{or} \qquad R_{TD} = R_{ad}. \qquad (2.13)$$

Equation (2.13) shows that the effect of resistance of the lead wires is cancelled. The resistance of the RTD is equal to the resistance of the adjustable resistor. The results of the temperature measurements using 3-wire RTD in the bridge are more precise than those using 2-wire RTD.

It has been shown that to obtain the accurate temperature measurements, the three wires of the RTD should have the same resistances. The resistances of the three wires can be maintained the same when they are made from the same material with the same size and length.

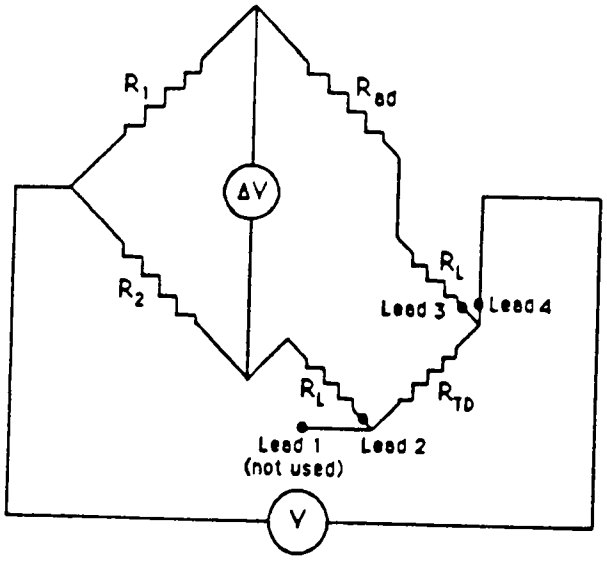
### ❖ Four-Wire RTD

A four-wire RTD is used when the connecting wires have different resistances and when the high accuracy temperature measurements are required. Two configurations of the Wheatstone bridge using a four-wire RTD are shown in Figure 2.8. In the first

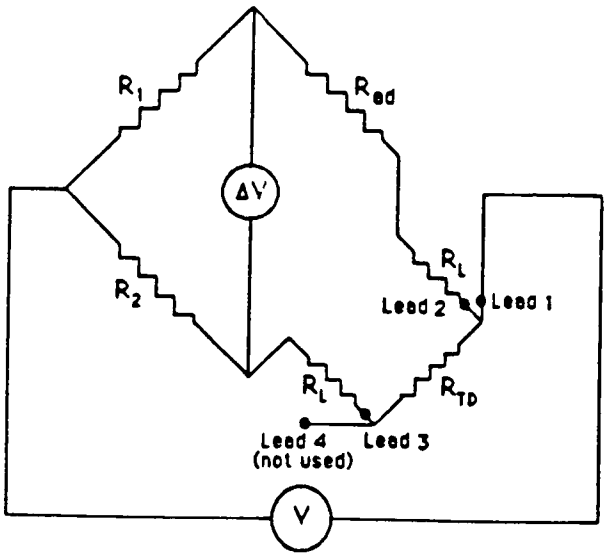



$R_1$ = resistor 1
$R_2$ = resistor 2
$R_{TD}$ = resistance temperature detector
$R_{ad}$ = adjustable resistor
$R_L$ = lead wire resistance
V = voltage

Figure 2.7. A Wheatstone Bridge With A Three-wire RTD Connection




(a) First Configuration




(b) Second Configuration

Figure 2.8. A Wheatstone Bridge With A Four-wire RTD Connection

configuration, the first lead of the RTD is not used. In the second configuration, the fourth lead is not used.

If the bridge is in balance, the first configuration yields

$$R_{S1} = R_{TD} + R_{L3} - R_{L2}\,, \tag{2.14}$$

and the second configuration yields

$$R_{S2} = R_{TD} + R_{L2} - R_{L3}\,. \tag{2.15}$$

One can obtained $R_{TD}$ by averaging Equation (2.14) and (2.15) as follows [7] :

$$R_{TD} = \frac{(R_{S1} + R_{S2})}{2}\,. \tag{2.16}$$

However this method requires rebalancing and lead reversal.

In industrial measurements, another method using a four-wire RTD configuration does not employ a Wheatstone bridge. The circuit of this method is shown in Figure 2.9. The two lead wires are connected to a constant current power supply, and the voltage across the other two lead wires is measured to determine the sensor resistance [7].

### ❒ SENSOR RESPONSE TIME

In temperature measurement, a temperature sensor is characterized by the time it takes to respond to a temperature change. In

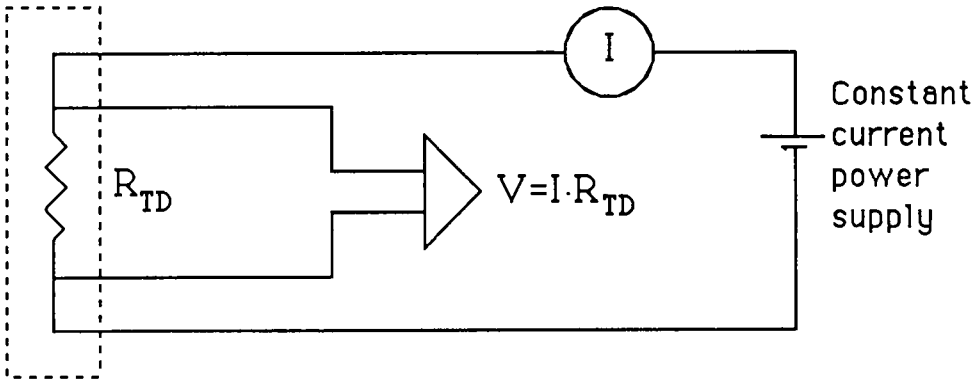



Figure 2.9. Circuit For A Four-wire RTD [7]

this research, the temperature transient used to characterize the temperature sensors is a step function. It is described as a function in which the temperature changes instantaneously to a new constant value.

### ❖ Time Constant

The time constant is used to characterize the response time of a first order system, which is represented with a first order differential equation. For a step function in input, the solution of a first order differential equation is

$$F(t) = F(\infty)\ (1 - e^{-t/\tau}), \qquad (2.17)$$

where

$F(t)$ = the system output at time t,

$F(\infty)$ = the final value of the output,

$\tau$ = the time constant of the system.

Since the time constant for a first order system is defined as the time required to achieve 63.2 percent of the change in temperature following a step change in input, if Equation (2.17) is evaluated at $t = \tau$, then

$$F(t = \tau) = 0.632\ F(\infty)\ . \qquad (2.18)$$

Even though the time constant has unambiguous meaning only for a first order system, it is used throughout this thesis to represent the

response time of an RTD to a step change in the surrounding temperature.

The time constant of a sensor, provided by sensor manufacturers, is commonly measured in a laboratory by plunging the sensor from air into flowing water (typically at 1 m/sec, at ambient pressure, and at room temperature or 85 $^{o}$C). This introduces a step change in temperature and generates a response transient similar to the one shown in Figure 2.10. The time constant of a sensor may change, because it is affected by sensor's environment and sensor's physical and thermal properties. In order to validate the performance of a sensor, its time constant must be measured in the environment where it is normally used.

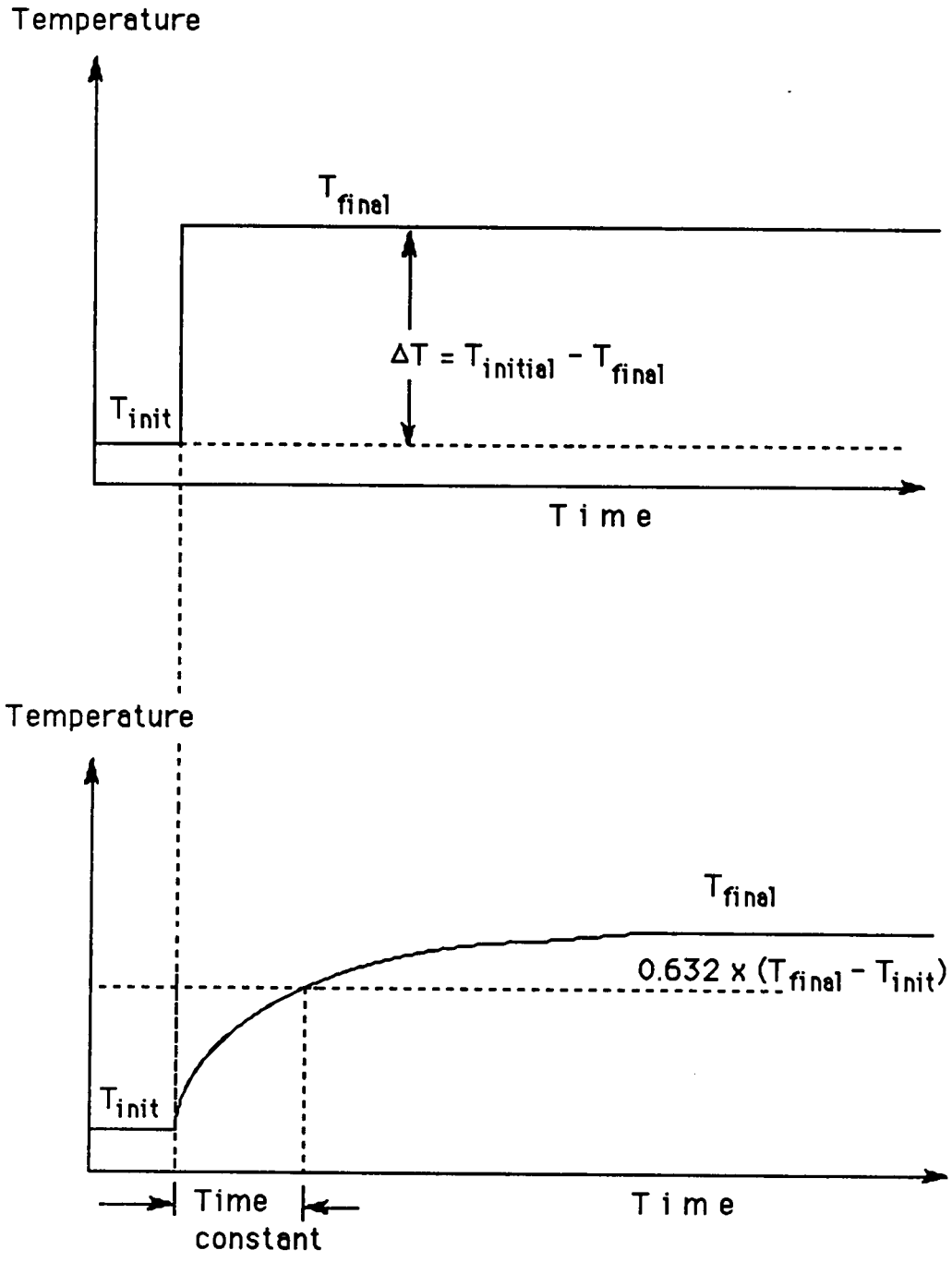



Figure 2.10. Time Constant Measurement From Plunge Test Transient

# CHAPTER 3
# CURRENT METHODS OF ESTIMATING THE TIME CONSTANTS OF RTDs

Since one of the important characteristics of RTDs is the response time, it is measured before the sensor is installed in a nuclear power plant. However, the sensor response time measured in a laboratory does not reflect the installed-RTD response time, because the response time is influenced by the environment where the sensor is installed. The response time of an installed RTD has to be measured regularly in-situ to ensure reactor safety. The method usually used is the loop current step response (LCSR) test.

## ❒ LCSR TEST

### ❖ Introduction

The LCSR test was developed in 1975 to measure the response time of thermocouples and RTDs in-situ [1]. This method can be applied to the response time measurements of installed RTDs without interfering with normal operation.

### ❖ Test Description

During normal operation, the current through the sensing element of an RTD is about 1 mA. In an LCSR test, the current is increased suddenly to about 50 mA to generate an internal step

change in heating current. The increased current induces heating at the sensing element, which is defined as joule heating, and generates a temperature transient, which rises slightly above the ambient temperature of the surrounding fluid. A typical LCSR test transient is shown in Figure 3.1.

However, the response transient for an external step change monitored in temperature is desired. An analytical transformation has been developed to convert the internal step response time to the external step response time of a sensor [2].

### ❖ LCSR Theory

The temperature response of sensing element to an internal step change in temperature is described by the following equation [7] :

$$T_{int}(t) = A_0 + A_1 e^{-t/\tau_1} + A_2 e^{-t/\tau_2} + A_3 e^{-t/\tau_3} + \ldots . \qquad (3.1)$$

The temperature response of sensing element to an external step change in temperature is given by the following equation [7] :

$$T_{ext}(t) = B_0 + B_1 e^{-t/\tau_1} + B_2 e^{-t/\tau_2} + B_3 e^{-t/\tau_3} + \ldots \qquad (3.2)$$

where

$t$ = time

$A_i$, $B_i$ = constants

$\tau_i$ = the modal time constants.

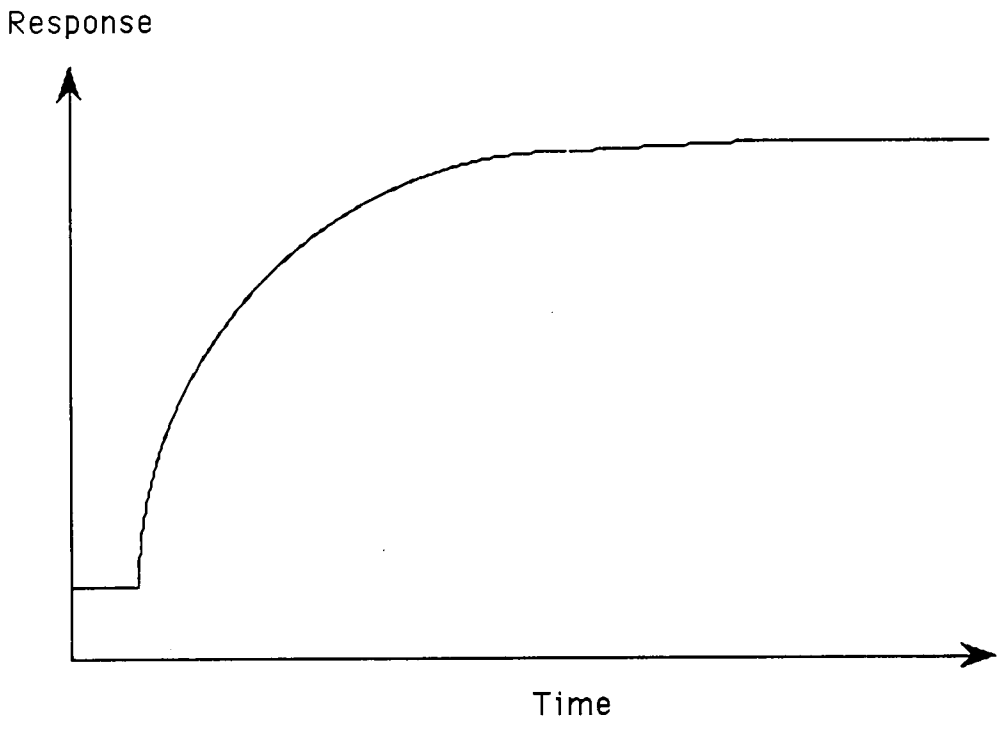



Figure 3.1. A Typical LCSR Transient

The two temperature responses are different from each other, because the $A_i$s from Equation (3.1) do not equal the $B_i$s from Equation (3.2), but they have the modal time constants, $\tau_i$, in common. Before analyzing the LCSR data, the following conditions must be shown to be satisfied:

- the heat transfer is predominantly one-dimensional, and
- the heat capacity of materials between the sensing element and the center of the sensor is insignificant [7].

The temperature response to an external step change in temperature, the response of interest, can be constructed if the constants, $B_i$, and the modal time constants, $\tau_i$, have been determined. The following relations are given to allow the calculation of $B_i$ [7] :

$$B_0 = \tau_1\tau_2\tau_3 \ldots \tag{3.3a}$$

$$B_1 = \frac{1}{(-1/\tau_1)\ (1/\tau_2 - 1/\tau_1)\ (1/\tau_3 - 1/\tau_1) \ldots} \tag{3.3b}$$

$$B_2 = \frac{1}{(-1/\tau_2)\ (1/\tau_1 - 1/\tau_2)\ (1/\tau_3 - 1/\tau_1) \ldots} \tag{3.3c}$$

$$B_3 = \frac{1}{(-1/\tau_3)\ (1/\tau_1 - 1/\tau_3)\ (1/\tau_2 - 1/\tau_3) \ldots} \tag{3.3d}$$

.

.

. .

The desired time constant is determined as follows :

$$\tau = \tau_1 \left[ 1 - \ln\left(1 - \frac{\tau_2}{\tau_1}\right) - \ln\left(1 - \frac{\tau_3}{\tau_1}\right) \ldots \right]. \qquad (3.4)$$

Practically, the response of interest may be constructed from $\tau_1$ and $\tau_2$ only, which are identified from LCSR data analysis.

However, the original application of the LCSR method to calculate the sensor time constant yielded errors of up to 26 percent, since the contribution of the higher modes to the overall time constant had not been accounted. Further analytical and numerical studies [4] revealed that information contained in the first two modes could be used to predict the contribution of the higher modes. This led to the development of a correction factor (CF), which was defined as follows:

$$CF = \frac{\text{overall time constant based on all modes}}{\text{overall time constant based on the first two modes}}$$

The application of the correction factor reduced the errors to within 10 percent.

### ❖ Limitations Of The Analytical Transformation

The analytical transformation has been developed using general nodal model for sensor heat transfer [2]. The application of this transformation is restricted by two factors [2,9]. The actual sensor heat transfer is assumed to be primarily one-dimensional.

But it is multi-dimensional, and therefore a knowledge of the first two modes is not sufficient for construction of the response to a fluid temperature step change. The sensing wire is also assumed to be located at the center of the RTD. When this is not always true, as in an industrial sensor, the analytical transformation is not valid to describe the heat transfer process.

### ❖ Experimental Procedures

Application of the LCSR test usually employs a standard Wheatstone bridge equipped with a switch as shown in Figure 3.2. This switch is used to increase current through the sensor, introducing a step change in temperature and generating joule heating in the sensing element. The joule heating then yields a temperature response. After it has been amplified, the response is recorded by a computer equipped with an analog to digital connector board. A timing circuit consisting of a battery and a switch is used to indicate the response starting time, which is recognized easily because the LCSR and timing circuit use the same switch.

Several LCSR tests have been performed in the laboratory of Department of Nuclear Engineering at The University of Tennessee, Knoxville (UTK). The following laboratory facilities are used in the LCSR test :

1. Two RTDs manufactured by the JMS Company
2. DC Power Supply of Hewlett Packard Model 6291A
3. Decade Resistor of General Radio USA Model 1433-F

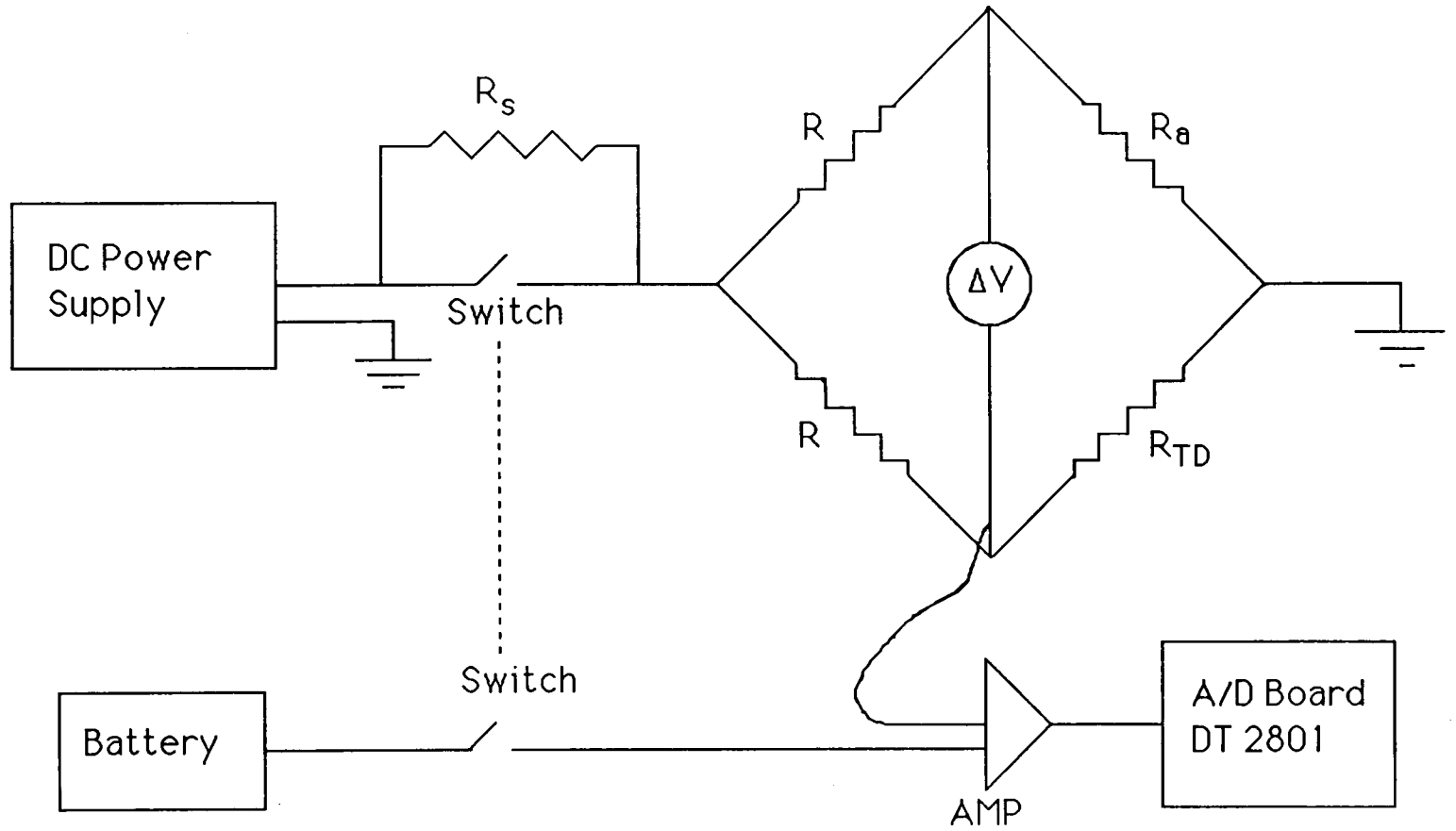



Figure 3.2. An Equipment Schematic For An LCSR Test

4. Wheatstone Bridge and Timing Circuit of Department of Nuclear Engineering, The University of Tennessee, Knoxville
5. Amplifier TEC Model 901
6. A/D Board DT 2801
7. Rotating Tank of Department of Nuclear Engineering, The University of Tennessee, Knoxville.

The specifications of the LCSR test are listed in Table 3.1. The procedure for performing an LCSR test is as follows :

1. The switch is opened allowing current about 1 mA through the RTD.
2. The bridge is balanced by adjusting the adjustable resistance (decade box).
3. The amplifier is adjusted to obtain signal amplification of 200.
4. The switch is closed to increase the current to about 20 mA through the RTD.
5. The LCSR test response transient is recorded by a PC computer equipped with A/D board DT2801.
6. The test is repeated three times.

## ❒ PLUNGE TEST

### ❖ Introduction

Plunge tests are performed in a laboratory to measure the time constant of a sensor, the time required to achieve 63.2 percent of

Table 3.1. LCSR Test Specifications

| Specification | Value |
|---|---|
| Power Supply | 4.2 volt |
| Current through the RTD at normal operation | 1.0 mA |
| Current increased | 20 mA |
| Amplification | 200 x |
| Sampling Frequency | 40 per second |
| Sampling Time | 25 seconds |
| High wattage $R_s$ | 2025 Ω |

the change in temperature following a step change in fluid temperature.

### ❖ Test Description

A plunge test is based on an external step change in temperature caused by a sudden immersion of a sensor into flowing water. The sensor operates at its normal sensing current, which is about 1 mA. A step change in temperature can be introduced either by plunging the sensor from room-temperature air into warm water or by plunging the warmed sensor into room-temperature water.

Figure 3.3 is an equipment schematic for a plunge test. The step change in temperature produces a response transient. The response transient is amplified before it is recorded in a computer. A timing circuit, consisting of a battery and a switch, is used to indicate the time at which the sensor enters the water. Using the conductivity of water, the switch is closed when the sensor touches water. The output is then amplified and recorded by a computer.

### ❖ Experimental Procedures

Several plunge tests have been performed in the laboratory of Department of Nuclear Engineering at The University of Tennessee-Knoxville. The laboratory facilities used in the plunge test include

1. Two RTDs manufactured by the JMS Company
2. DC Power Supply of Hewlett Packard Model 6291A
3. Decade Resistor of General Radio USA Model 1433-F

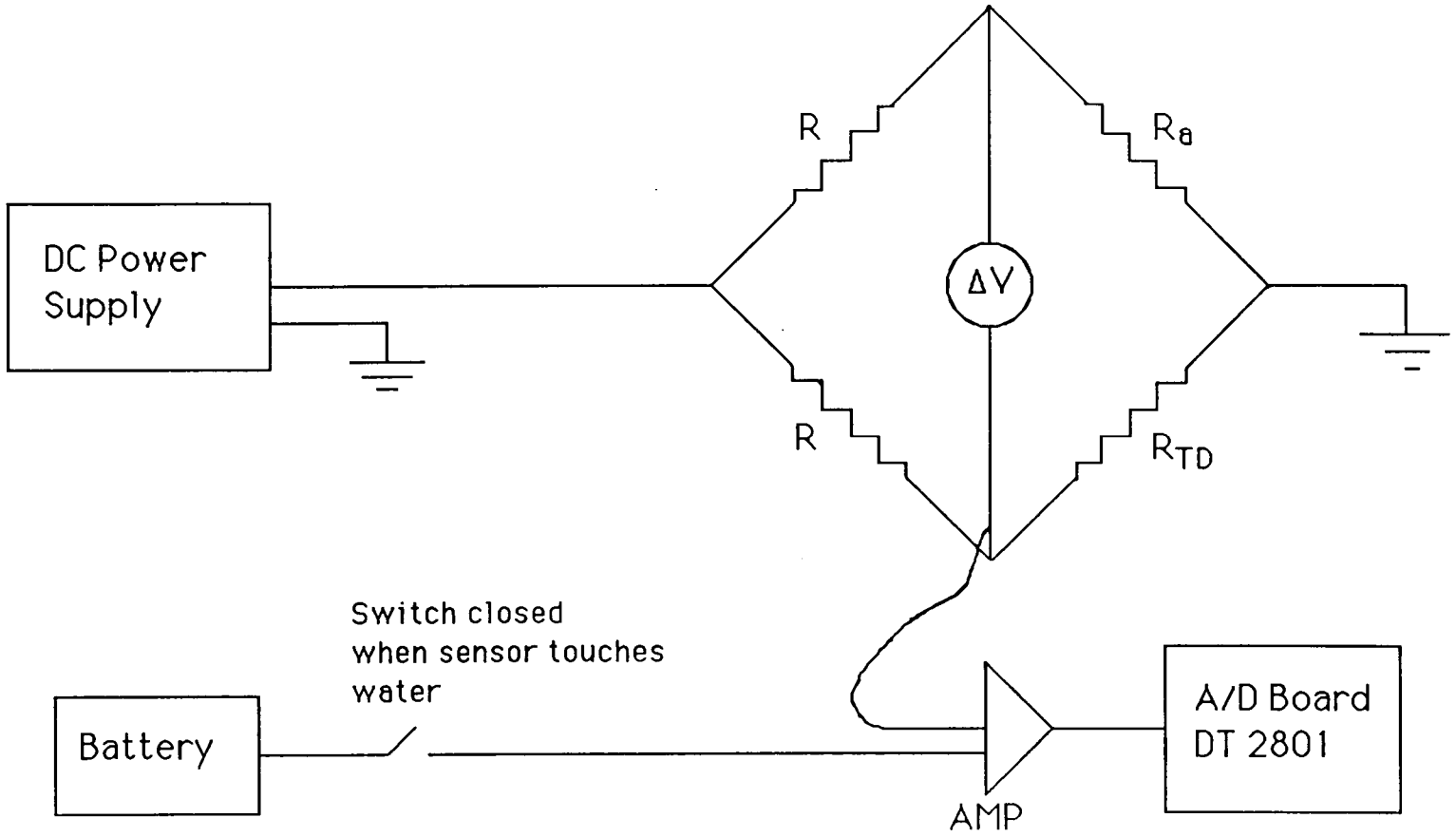



Figure 3.3. An Equipment Schematic For A Plunge Test

4. Wheatstone Bridge Circuit of Department of Nuclear Engineering, UTK
5. Amplifier TEC Model 901
6. A/D Board DT 2801
7. Rotating Tank of Department of Nuclear Engineering, UTK.

In a plunge test, the RTD is held in air before being plunged into the rotating tank of water in a location where a desired water velocity is maintained. Test specifications are listed in Table 3.2. The procedure for performing a plunge test is as follows:

1. The tank, which contains approximately 15 gallons of water, is rotated.
2. The RTD is mounted to the shaft of a pneumatic piston positioned above a region of water flowing at about 3 feet/sec.
3. The blower is turned on to warm the RTD.
4. The equipment schematic is configured as shown in Figure 3.4.
5. The bridge is balanced by adjusting the adjustable resistance (decade box).
6. The amplifier is adjusted to obtain an amplification of 2000.
4. The RTD is inserted into the flowing water.
5. The plunge test response transient is recorded by using a PCLAB equipped with the A/D board DT2801.

Several plunge tests have been performed on two RTDs. As mentioned in Chapter 2, the sensor time constant may change, because the sensor response time is affected by its environment and

Table 3.2. Plunge Test Specifications

| Specification | Value |
|---|---|
| Power Supply | 4.2 volt |
| Current through the RTD at normal operation | 1.0 mA |
| Amplification | 2000 x |
| Sampling Frequency | 20 per second |
| Sampling Time | 50 seconds |

its physical and thermal properties. To simulate several RTD's with different response times, the sheath of each RTD was wrapped with electrical tape altering the time constants. Those time constants would be used as desired output in training neural networks.

## ❒ SAMPLING DATA

The data required to train the networks are obtained from several sets of LCSR and plunge test data. Loop current step response and plunge test specifications are listed in Table 3.1 and Table 3.2. The equipment schematics for an LCSR test and a plunge test are shown in Figure 3.2 and 3.3.

There were two sets of LCSR tests. The first set, simulating high-quality equipment, used a Wheatstone bridge consisting of 2 high-wattage resistors, an adjustable resistor, and an RTD. The high-wattage resistors reduce resistance changes due to an increased current through the bridge. The second set of LCSR tests used a Wheatstone bridge employing 2 low-wattage resistors, an adjustable resistor, and an RTD to simulate inferior equipment. The resistances of the low-wattage resistors will change while the current through the bridge is increased. This produces a different measurement of the sensor response time from those in the first set of tests. The tests with low-wattage resistors are used to determine the neural network's ability to handle equipment defficiencies.

The plunge tests were used to measure the RTDs' time constants. Those time constants are listed in Table 3.3 and Table 3.4.

Data from the LCSR and plunge tests were sampled by using three interface program: WDSAMPLE, WTIMEPLT, and MATCON programs. Data in the file created by WDSAMPLE program is raw data. This data file has to be converted to a readable data file (ASCII file) using MATCON program. The readable data file allows the time constant of an RTD to be calculated from plunge test data or data required in training neural networks to be selected. Figure 3.4 shows those steps. WTIMEPLT program is used to plot the response transient of either the LCSR test or plunge test. Examples of those executable programs are described in Appendix D.

Table 3.3. The Results Of Plunge Tests On RTD A

| Experiment No. | Time Constant, $\tau$ |
|---|---|
| 1 | 4.03 |
| 2 | 4.12 |
| 3 | 4.21 |
| 4 | 4.23 |
| 5 | 4.52 |
| 6 | 4.89 |
| 7 | 4.90 |
| 8 | 4.93 |
| 9 | 5.05 |
| 10 | 5.30 |
| 11 | 5.33 |
| 12 | 5.44 |
| 13 | 6.48 |
| 14 | 6.57 |
| 15 | 6.78 |
| 16 | 6.95 |
| 17 | 7.02 |
| 18 | 7.20 |
| 19 | 7.83 |
| 20 | 7.96 |
| 21 | 8.10 |

Table 3.4. The Results Of Plunge Tests On RTD B

| Experiment No. | Time Constant, $\tau$ |
|---|---|
| 1 | 3.83 |
| 2 | 3.88 |
| 3 | 3.95 |
| 4 | 4.26 |
| 5 | 4.45 |
| 6 | 4.76 |
| 7 | 4.91 |
| 8 | 5.71 |
| 9 | 6.19 |
| 10 | 6.50 |
| 11 | 7.03 |
| 12 | 7.13 |
| 13 | 7.33 |
| 14 | 7.42 |
| 15 | 7.48 |
| 16 | 7.60 |
| 17 | 7.65 |
| 18 | 7.70 |
| 19 | 7.74 |

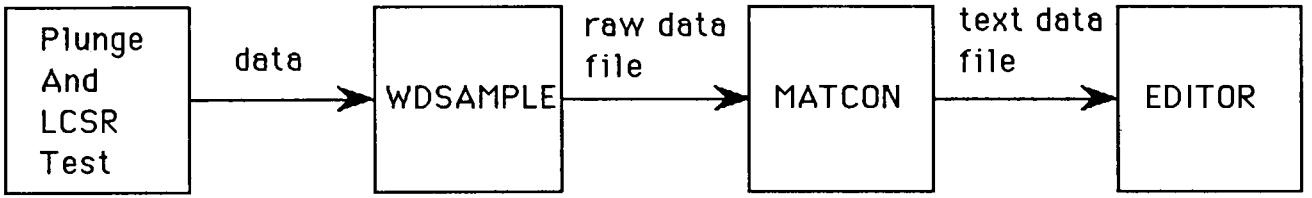



Figure 3.4. Steps To Obtain Text Data File

# CHAPTER 4
# NEURAL NETWORKS

## ❐ INTRODUCTION

Neural networks have been applied in a number of areas. One area that neural networks excel to deal with is pattern recognition. In this thesis, neural networks are used to solve a pattern recognition problem that is prediction of sensor's time constant from LCSR test transient.

A neural network is an information processing system [19], inspired by the human brain. It consists of a number of simple and highly interconnected processing elements (PEs), each of which processes information by its dynamic state response to external input. A PE is a simple device that can receive a number of input signals as shown in Figure 4.1. The PE translates the input signals into a single output signal, which is then sent to other PEs as input signals through the connections between PEs. Each connection is characterized by a weight.

The translation of the input signal to the output signal in a PE consists of three steps. First the PE sums all inputs it is receiving. The summed input, $I_j$, is determined by multiplying each input signal by the weight of that connection:

$$I_j = \sum_{i=1}^{n} x_i w_{ij} \qquad (4.1)$$

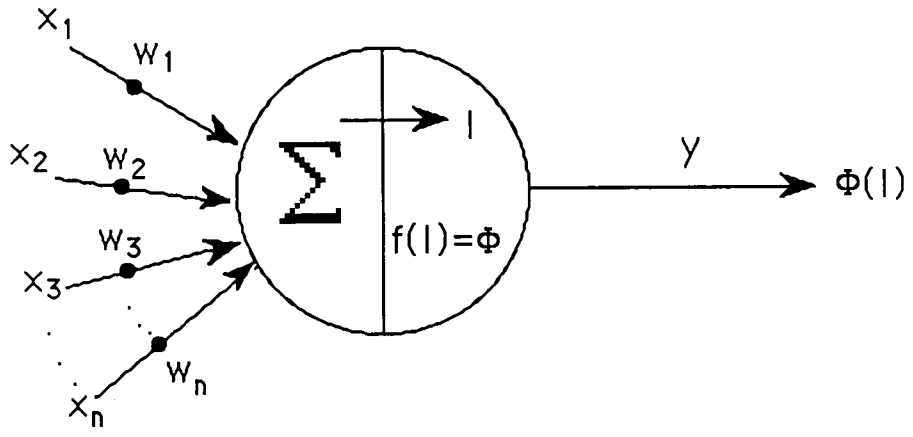



Figure 4.1. A Processing Element's Operation

where

$I_j$ = the summed input, sometimes called the net input, received by PE j from a total of n PEs in the network,

$x_i$ = the input signal from the PE i,

$w_{ij}$ = the weight on the connection from PE i to PE j.

Second, the summed input is converted to a PE's activation level by using an activation function. The third step is converting the PE's activation level to an output signal.

### ❖ Neural Network Operations

Most neural networks consists of three layers: one input layer, one hidden layer, and one output layer. An example of a three-layer network is shown in Figure 4.2. In a neural network operation, each input-layer PE receives a part of the input pattern. The input pattern is transmitted to the hidden-layer PEs after it has been modified by the weights in the connections between input layer and hidden layer. Since these weights have different value for each hidden-layer PE, it receives a variety of input patterns, resulting a variety of output responses [19].

The hidden-layer output responses are modified by the weights in the connection between hidden layer and output layer and are then transmitted to the output-layer PEs. The pattern of output-layer PE responses is the network's response to the original input pattern [19].

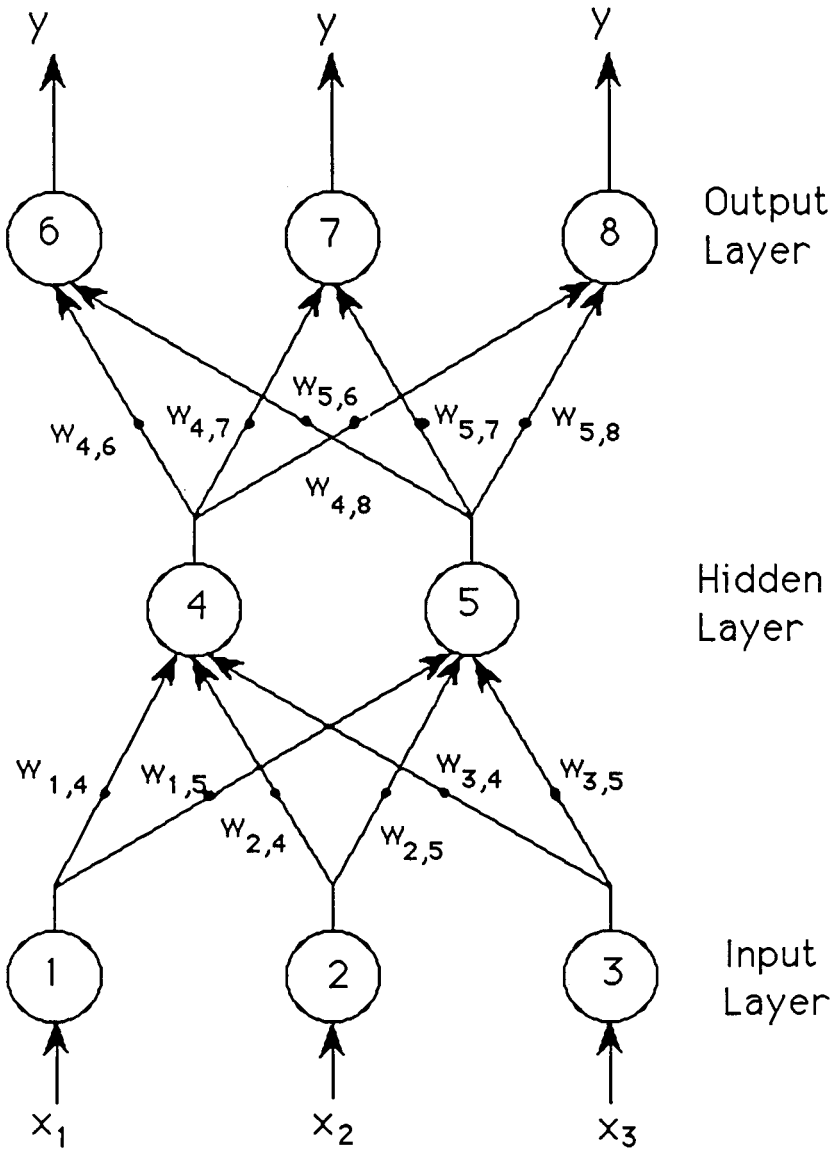



Figure 4.2. An Example Of A Three-layer Neural Network Architecture

## ❖ Training in Neural Networks

Neural networks solve a problem by modifying the weights in the network. This process is called training. One of the common training techniques is supervised training. In it, the network is trained with an input pattern and a corresponding desired output pattern. The supervised training technique employed by the neural networks in this research propagates output errors from the output layer to the input layer. This neural network is often called a back-propagation network.

## ❒ BACKPROPAGATION NEURAL NETWORKS

This thesis will use the term backpropagation network to represent a network implementing backpropagation of errors. The backpropagation network has at least three layers: one input layer, one or more hidden layers, and one output layer. An example of a three-layer backpropagation network is shown in Figure 4.3. Backpropagation networks implement a learning rule called generalized delta learning rule. It is a gradient descent system to compute the weights that minimize the total squared output error. Figure 4.4 shows the weight changes in the generalized delta rule.

In translating an input signal to an output signal, a backpro-pagation network uses an activation function to convert a net input signal to an activation level. An activation function used by back-propagation networks is a sigmoid function. The sigmoid function is expressed by an S-shaped curve as shown in Figure 4.5 and may be

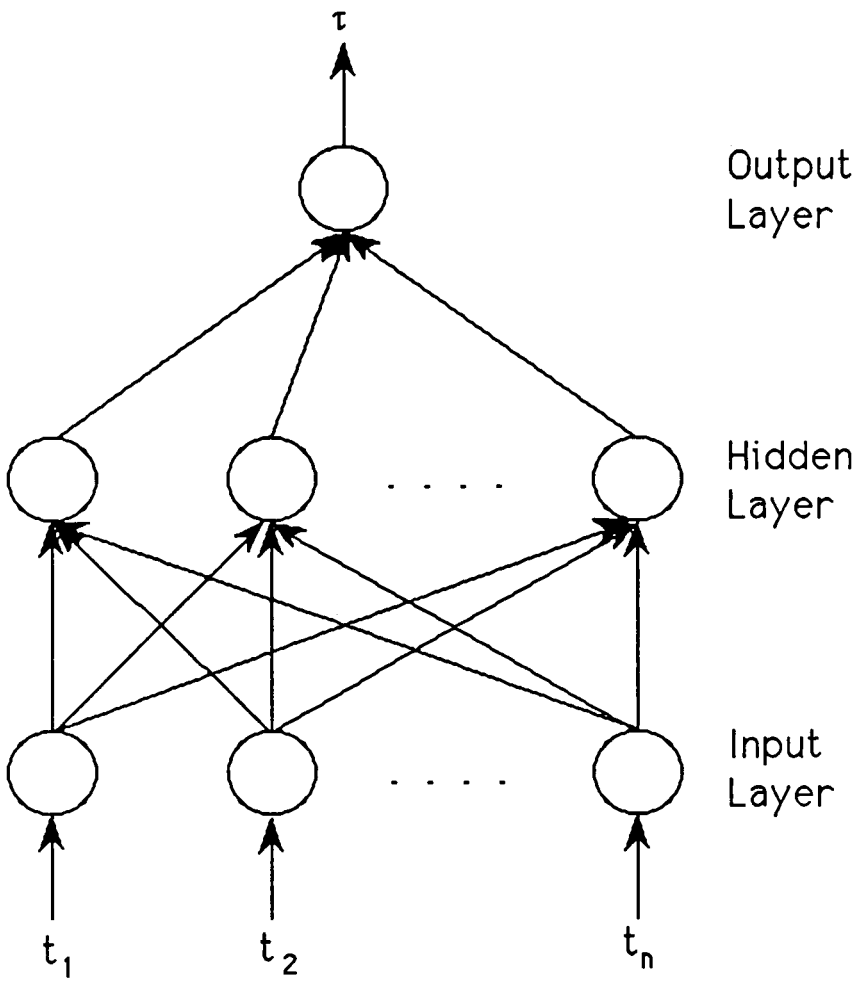



Figure 4.3. A Typical Three-layer Backpropagation Network For Time Constant Prediction

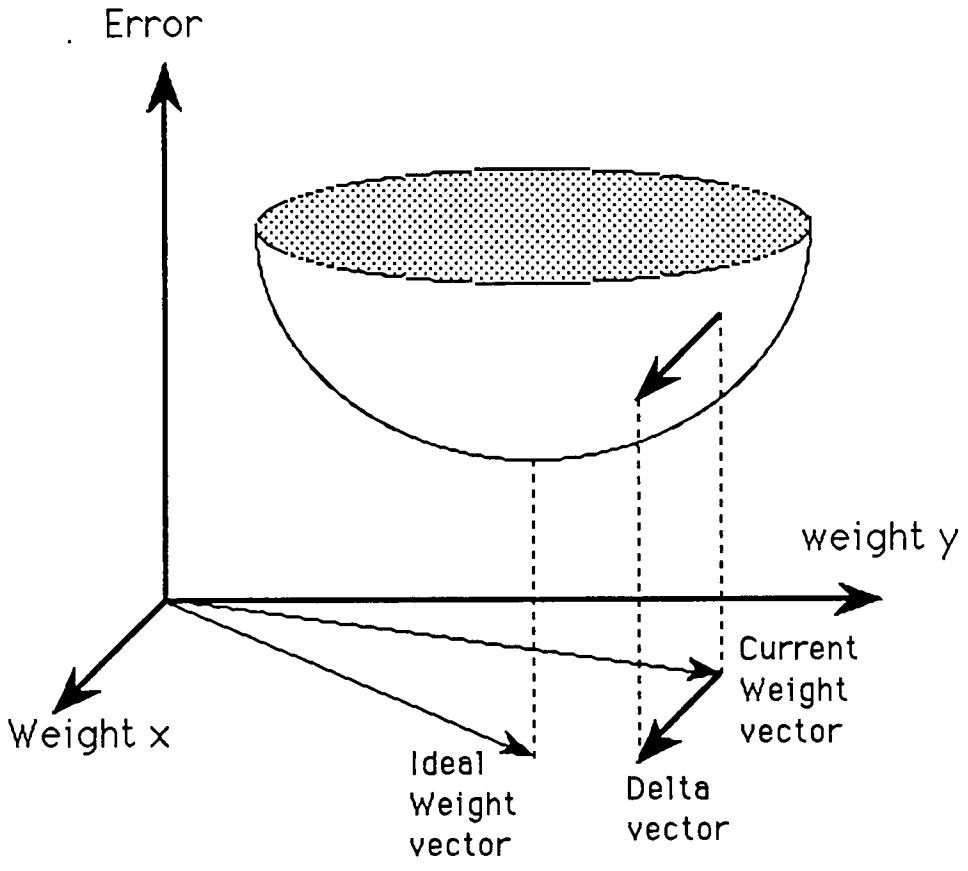



Figure 4.4. Weight Changes In The Generalized Delta Rule

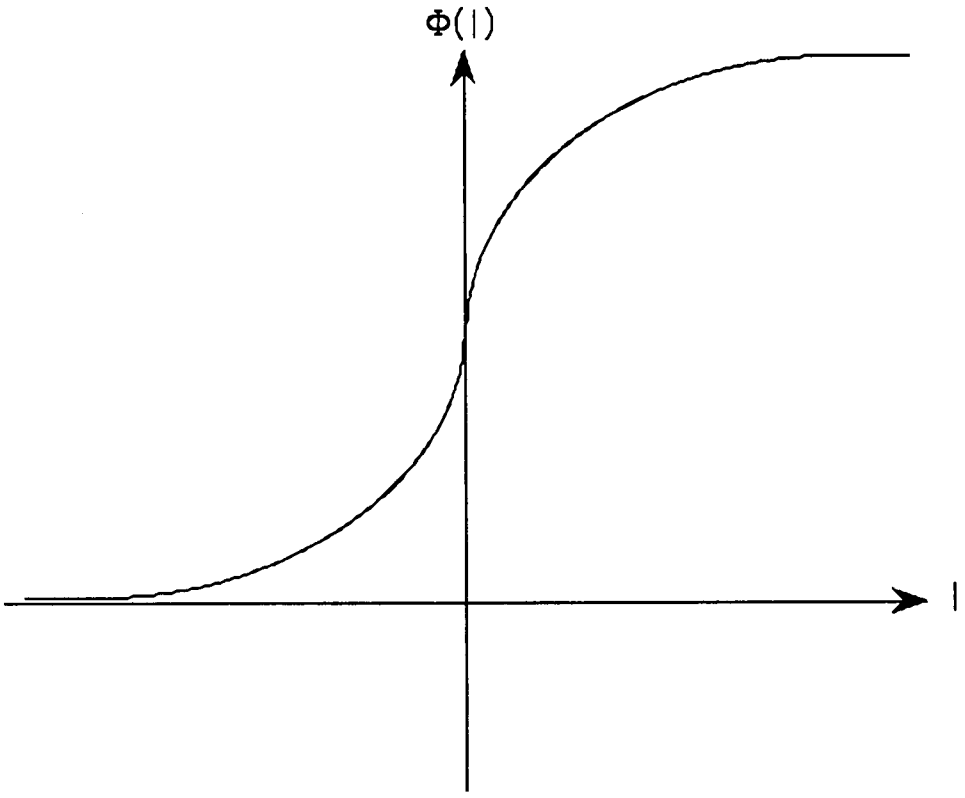



Figure 4.5. A Sigmoid Function Curve

represented by

$$\Phi(I) = \frac{1}{1 + e^{-I}} \cdot \qquad (4.2)$$

The value of $\Phi(I)$ is 0.0 as the net input I approaches negative infinity, and is 1.0 as I approaches positive infinity. The sigmoid function provides control on the change of the weights. The useful property of this activation function is that its derivative $\Phi'(I)$, shown in Figure 4.6, is always positive and continuous as described by the following equation [19] :

$$\Phi'(I) = \Phi(I)\ [1-\Phi(I)] \ . \qquad (4.3)$$

This derivative is used to modify the output-layer weights.

### ❖ Training Backpropagation Neural networks

During training, a backpropagation network operates in two step procedures: forward and backward activity. This two-step procedure is illustrated in Figure 4.7. Forward activity is initiated by randomizing the weights on the network interconnections to assure that the network is not saturated by large value of weights. Next a training pair is selected from the training set. The training pair consists of an input pattern and a desired output pattern. The input pattern is applied to the input layer generating activity in the input-layer PEs. This activity propagates forward through each of

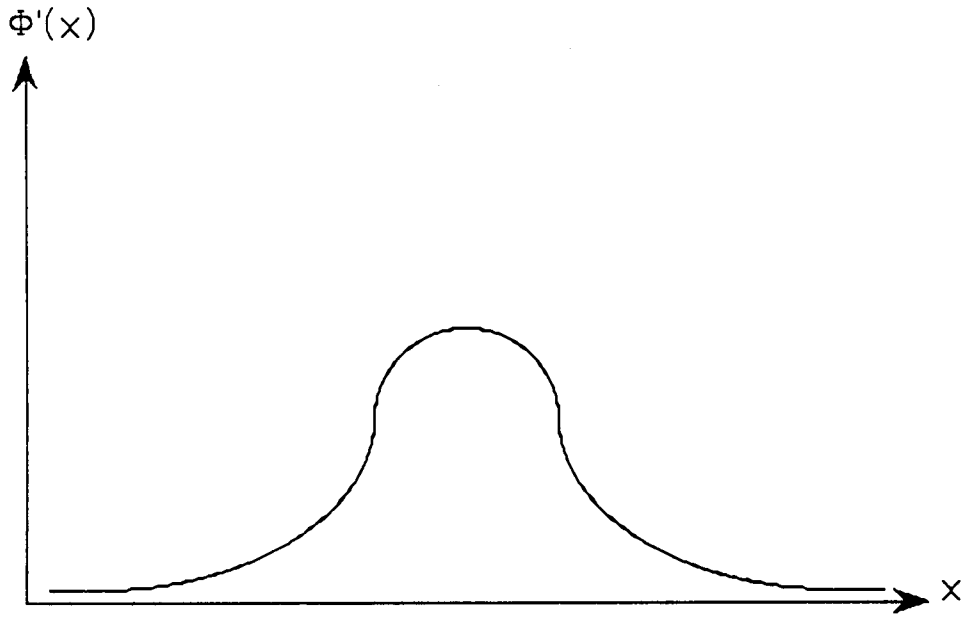



Figure 4.6. A Curve Of A Sigmoid Function Derivative

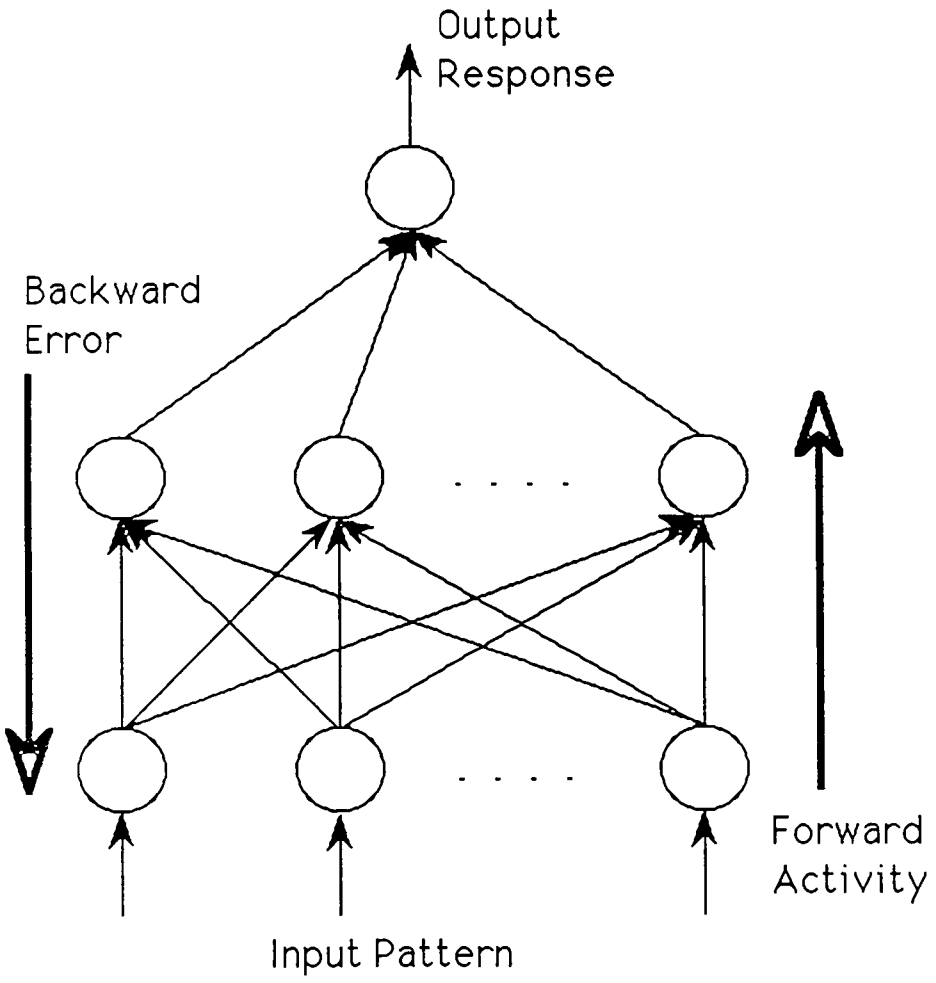



Figure 4.7. Two Step Procedures Of Training In A Backpropagation Network

the layers of the network until the output layer produces an output pattern. Backward activity is started by comparing the desired output pattern with the network outtput pattern. The comparison results errors that are backpropagated through the layers of the network, modifying the weight of each layer.

**Modifying The Output-Layer Weights**

For a single weight, $w_{pq}$, from a single PE p in the hidden layer j to the PE q in the output layer k, as shown in Figure 4.8. The output-layer weight can be modified in several steps.

1. The output error signal of the PE q in the output layer k is computed by the following equation [20]:

$$e_k = [\, T_k - \Phi_k(I) \,] \,, \tag{4.4}$$

where

$T_k$ = the target in the output layer k

$\Phi_k(I)$ = the sigmoid function in the output layer k.

2. The error signal is multiplied by the derivative of the activation function of that output-layer PE. This yields [20]

$$\delta_{qk} = e_k \Phi'_k(I)$$

$$= [\, T_k - \Phi_k(I) \,] \, [\Phi_k(I) \; (1-\Phi_k(I))] \;. \tag{4.5}$$

3. The change in the weights $w_{pq}$ is computed by multiplying $\delta_{qk}$ with the activation function of the PE in hidden layer j, $\Phi_k(I)$:

$$\Delta w_{pq,k} = \beta \, \Phi_j(I) \, \delta_{qk} \;. \tag{4.6}$$

where

$\beta$ = the learning rate,

$\Phi_j(I)$ = the sigmoid function in the hidden layer j.

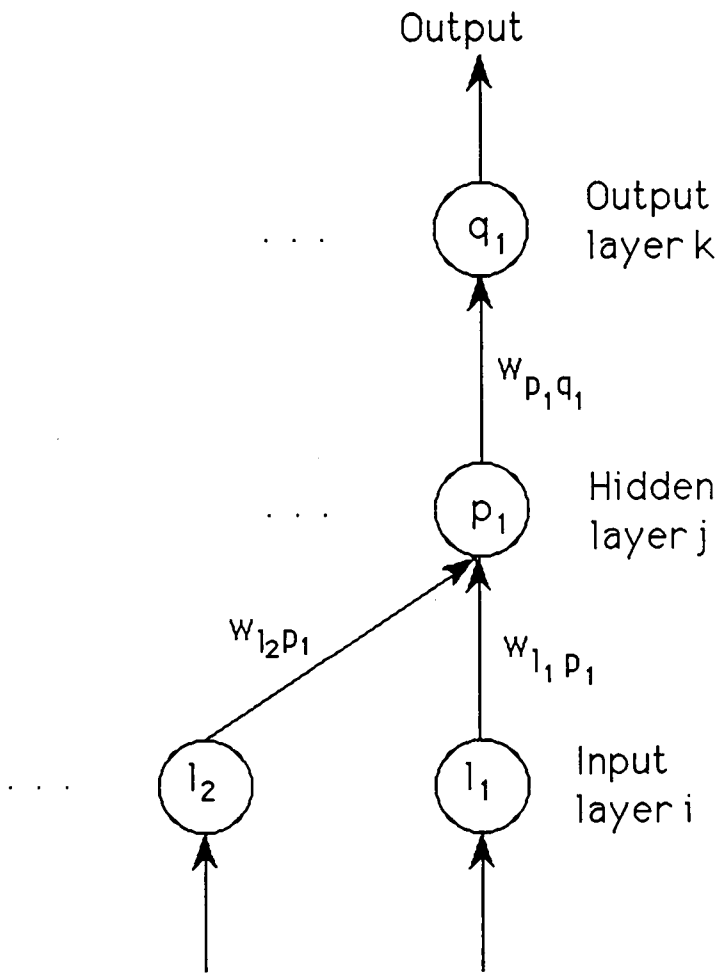



Figure 4.8. Schematic For Modifying An Output-layer Weight

4. The updated weight of $w_{pq}$ is determined from [20]

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k} \quad . \tag{4.7}$$

5. The other output-layer weights are computed in the same manner.

**Modifying The Hidden-Layer Weights**

Since hidden layers have no target output, hidden layers are trained by backpropagating the error through the network to modify the weights at each layer. The equations used for modifying the hidden-layer weights are the same as for modifying the output-layer weights, but $\delta$ is generated without a target output.

For a single hidden-layer PE that is just below the output layer, as shown in Figure 4.9 [20], the $\delta_{pj}$ is obtained by summing all such products and multiplying by the derivative of the sigmoid function as follows [20] :

$$\delta_{pj} = \Phi_{p1,j}(I)\ [1-\Phi_{p1,j}(I)]\ [\ \Sigma\delta_{qk}\ w_{pq,k}] \ , \tag{4.8}$$

where $\Phi_{p1,j}(I)$ is the output from PE $p_1$ in hidden layer j, and $w_{pq,k}$ is computed by using Equation (4.9) and $\delta_{qk}$ is obtained from Equation (4.7). The change in hidden-layer weights is then calculated by [20]

$$\Delta w_{Ip,j} = \beta\ \Phi_{I1,j}(I)\ \delta_{pj}(I) \ , \tag{4.9}$$

where $\Phi_{I1,j}(I)$ is the output from PE in input layer $I_1$ to hidden layer j. The value of $w_{Ip,j}$.is determined from the following equation:

$$w_{Ip,j}(n+1) = w_{Ip,j}(n) + \Delta w_{Ip,j} \quad . \tag{4.10}$$

The other hidden-layer weights are computed in the same way

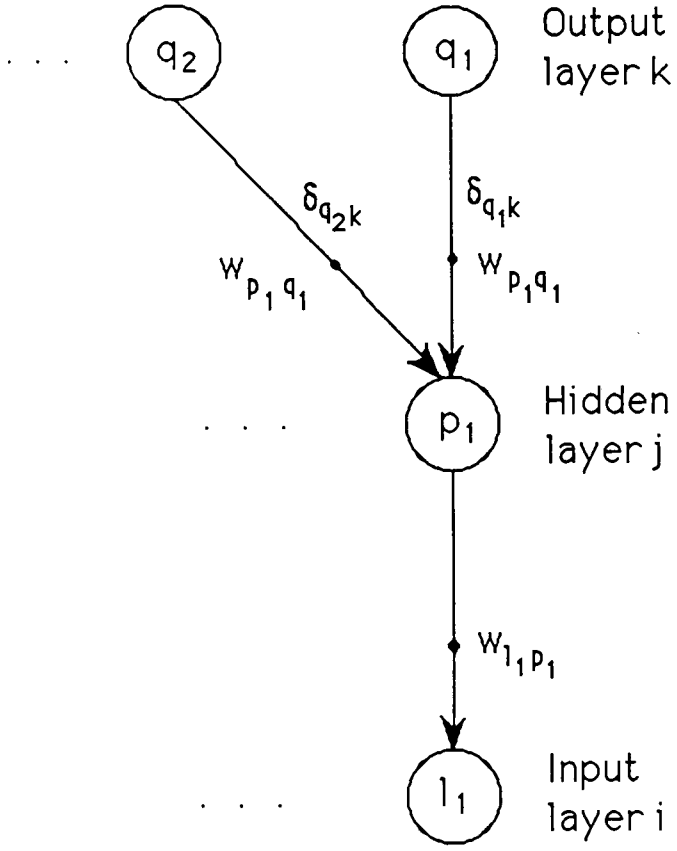



Figure 4.9. Schematic For Modifying A Hidden-layer Weight

# CHAPTER 5
# OPTIMIZING BACKPROPAGATION NETWORKS

## ❒ INTRODUCTION

The backpropagation network is used in many fields because it is simple to implement and works well for various application. However, it has some weaknesses. It often takes a long time to train because the network is trained in two procedures: forward and backward activity. When the network is trapped in a local minimum error, the error of the output response does not converge to a given tolerable error.

These problems can be avoided by adding a momentum term to the generalized delta rule, selecting the optimum number of layers, and selecting the optimum number of PEs in each layer.

## ❒ MOMENTUM TERM

A momentum term is added to the generalized delta rule to avoid trapping the network in a local minimum error. The generalized delta rule is now defined as

$$\Delta w_{ij} = \beta E \Phi(I) + \alpha \Delta w_{ij}^{previous} , \qquad (5.1)$$

where

$\beta$ = the learning rate,

E = the output error,

$\Phi(I)$ = the activation function,

$\alpha$ = the momentum term, between 0.0 and 1.0 .

Equation (5.1) indicates that $\Delta w_{ij}$ is always positive. The momentum term maintains the weight vector moving in the same direction, so the weight vector does not easily get trapped in a local minimum.

## ❒ SELECTING THE OPTIMUM NUMBER OF LAYERS

Generally, a backpropagation network consists of one input layer, one output layer, and one or more hidden layer. There is no agreement on how to choose the right number of hidden layers. Interpreting Kolmogorov's theorem, Hecht-Nielsen (1987) [21] proved that a network with one hidden layer can compute any arbitrary input function. To compute a classification problem, Cybenko (1988) [21] showed that one hidden layer is enough. He also showed that an arbitrary output function of the inputs can be calculated sufficiently by using two hidden layers. Meanwhile, Lippman [21] stated that two hidden layers are sufficient to compute any classification problem. Maren's advice is to use one hidden layer for classification problem, and two hidden layers if the output is a continuous function of the input [21].

## ❒ SELECTING THE OPTIMUM NUMBER OF PEs

The number of PEs in each layer of a backpropagation neural network is very crucial since those PEs have a great influence on the network performance and training time. The number of output-layer

PEs depends on the number of outputs required. In this measurement of temperature response time, the networks use one output-layer PE because the networks predict only one time constant from a set of LCSR data.

Selecting the optimum number of hidden-layer PEs is difficult. If the number of hidden-layer PEs is too small, the network's output error may not converge to an acceptable error during training. On the other hand, if the number of hidden-layer PEs is too large, the network will fail to recognize the features in the input patterns [21]. There is no formula that assures the exact number of hidden-layer PEs needed by a network has been determined, but some methods have been developed to select the number of hidden-layer PEs. Kolmogorov [21] adviced approximating the number of hidden-layer PEs using

$$H = (2N+1), \tag{5.2}$$

where

H = the number of hidden-layer PEs,

N = the number of input-layer PEs.

Meanwhile Eryurek [22] indicated that the number of hidden-layer PEs can be estimated by

$$H = (N \log_2 I \pm N), \tag{5.3}$$

where I is the number of input/output data sets. Both relationships have been used to train the networks with the difference in the second one. The results of the training are discussed in Chapter 7.

Usually, the number of input-layer PEs is equal to the number of samples that adequately represent the transient.

# CHAPTER 6
# NEURAL NETWORK METHODS

## ❒ INTRODUCTION

Neural networks have been used widely in many tasks. In this research for the measurement of the time constant of a temperature sensor, a neural network is applied to eliminate the difficulties of the analytical transforma-tion that converts internal response transients to desired external response transients. A backpropaga-tion network is implemented to predict the time constants of RTDs. The neural-network software package used in this research is selected from an evaluation of two neural-network software packages: NeuralWorks and ANSim.

## ❒ NEURAL NETWORK SOFTWARE EVALUATION

Neural-network training sets have been prepared to evaluate the performance of the networks and to compare the convergence results of the two software packages.

NeuralWorks was developed by NeuralWare. It employs a pro-cessing element that allows the creation of a wide variety of neural networks using the same data structures. The advantage is four fold. First, by keeping much of data structures in common, only a very small portion of the program changes for different network types. Second, this model makes it possible to add new features to NeuralWorks that immediately apply to all network types without

the necessity of modifying each individual one. Third, this architecture readily lends itself to multiple parallel processors. Fourth, it provides the user with the maximum flexibility in designing new network types and variations on existing networks.

ANSim (Artificial Neural System Simulation) program, developed by the Science Applications International Corporation (SAIC), is integrated under Microsoft Windows to provide an effective, easy-to-use interface. ANSim can be used to configure any number of ANS networks. It drives each network with a sequence of training and/or input data. For each model, ANSim will monitor the response, capture the output, and save the configuration for later re-use. The speed at which ANSim will process a neural network depends on the particular hardware configuration, network paradigm, selection of network architecture parameters, and display options enabled.

### ❖ Comparison Between Neural Works and ANSim

As mentioned in Chapter 4, the generalized delta rule guarantees gradient descent in the total root mean square (RMS) error. This error is computed by summing the square of the target minus the output for every output unit and for every pattern, averaging that result and then taking the square-root. The total RMS error is

$$\text{Total RMS Error} = \sqrt{\frac{\sum_{p=1}^{n}\sum_{i=1}^{n}(t_{p,i}-o_{p,i})^2}{\#\text{ patterns x }\#\text{ output units}}}, \quad (6.1)$$

where

p = the pattern,

i = the output unit,

t = the target output unit,

o = the desired output unit.

Neural Works and ANSim implement the generalized delta rule differently. Neural Works uses the generalized delta rule to update the connection weights after each pattern vector is processed. ANSim applies the delta rule only after the complete training set has been introduced to the network, and the accumulative error of each vector is used to update the connection weights. The method in ANSim requires less time to process each vector, but it may increase the number of training cycles needed to converge.

### ❖ Training Sets

Several training sets were developed to select between Neural Works and ANSim. Three-layer networks implementing backpropagation paradigm were trained using simulated RTD temperature-response data to predict RTD's time constant. The simulated data was used to test the neural network method before acquiring costly sensors. Simulated responses of five RTD's with time constants between 0.1 seconds and 0.5 seconds were used for training sets.

The size of the input vector determines the number of input-layer PEs required by the network. The number of hidden-layer PEs is approximated by a formula:

$$H = N \log_2 I , \qquad (6.2)$$

where

H = the number of hidden-layer PEs,

N = the number of input-layer PEs,

I = the number of training input patterns.

The formula is a simplified form of the Equation (5.3).

The networks were trained by NeuralWorks and ANSim under the same number of cycles. The results of the training sets, shown in Figure 6.1, indicate that use of ANSim results is lower average relative errors than those of NeuralWorks: ANSim will be used for application of neural networks to this measurement of temperature sensor response time.

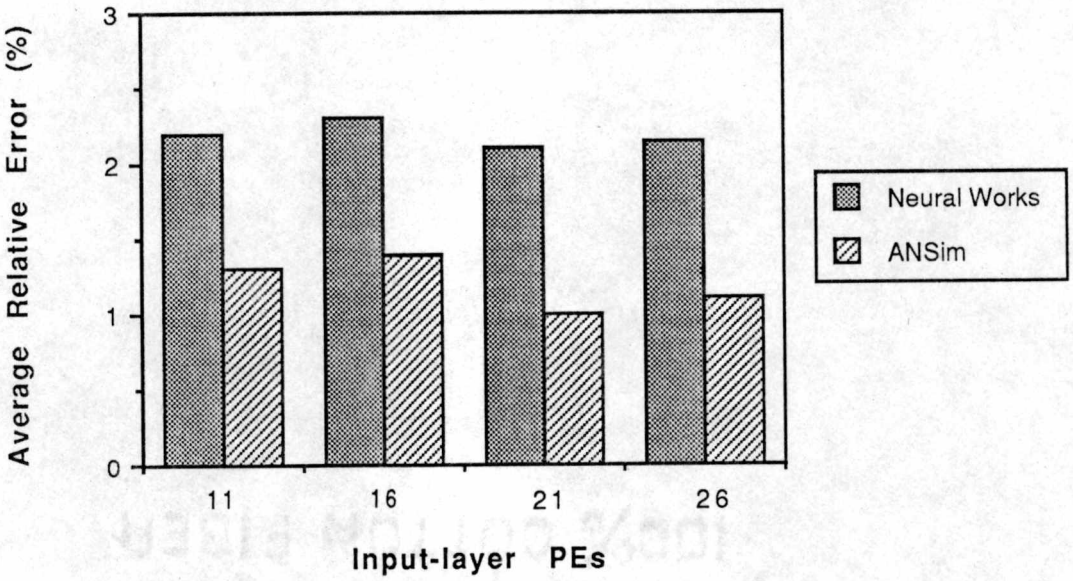



Figure 6.1. Comparison Study Between NeuralWorks And ANSim

# CHAPTER 7
# RESULTS

## ❒ TRAINING NEURAL NETWORKS

As explained in Chapter 5, training is achieved by modifying the weights in the network interconnections until the network's output results are within an acceptable error. The training procedure is illustrated in Figure 7.1. Sets of input/output data are used to train the networks. The input data are selected from the LCSR transients at equal intervals between 1 and 12 seconds. The desired output is the time constant obtained from the plunge test. The time constants used as the desired output are listed in Table 3.3 and 3.4. The input and output data are normalized to be between -0.5 and 0.5. Normalization removes the mean (i.e., offset the mean to zero) and scales data within a specific range. It ensures that the data in each file is compatible with the ANSim computational methodology [24].

Several networks, consisting of three layers: an input layer, a hidden layer, and an output layer, have been trained by the ANSim software. The neural network specifications are shown in Table 7.1. The number of input-layer processing elements (PEs), which represents the number of samples selected from LCSR transient, are 20, 40 and 60. These values were chosen arbitrarily. Examples of 20, 40, and 60 input data are shown in Figure 7.2, Figure 7.3, and Figure 7.4, respectively. The output-layer PE corresponds to RTD's time constant obtained from the plunge test.

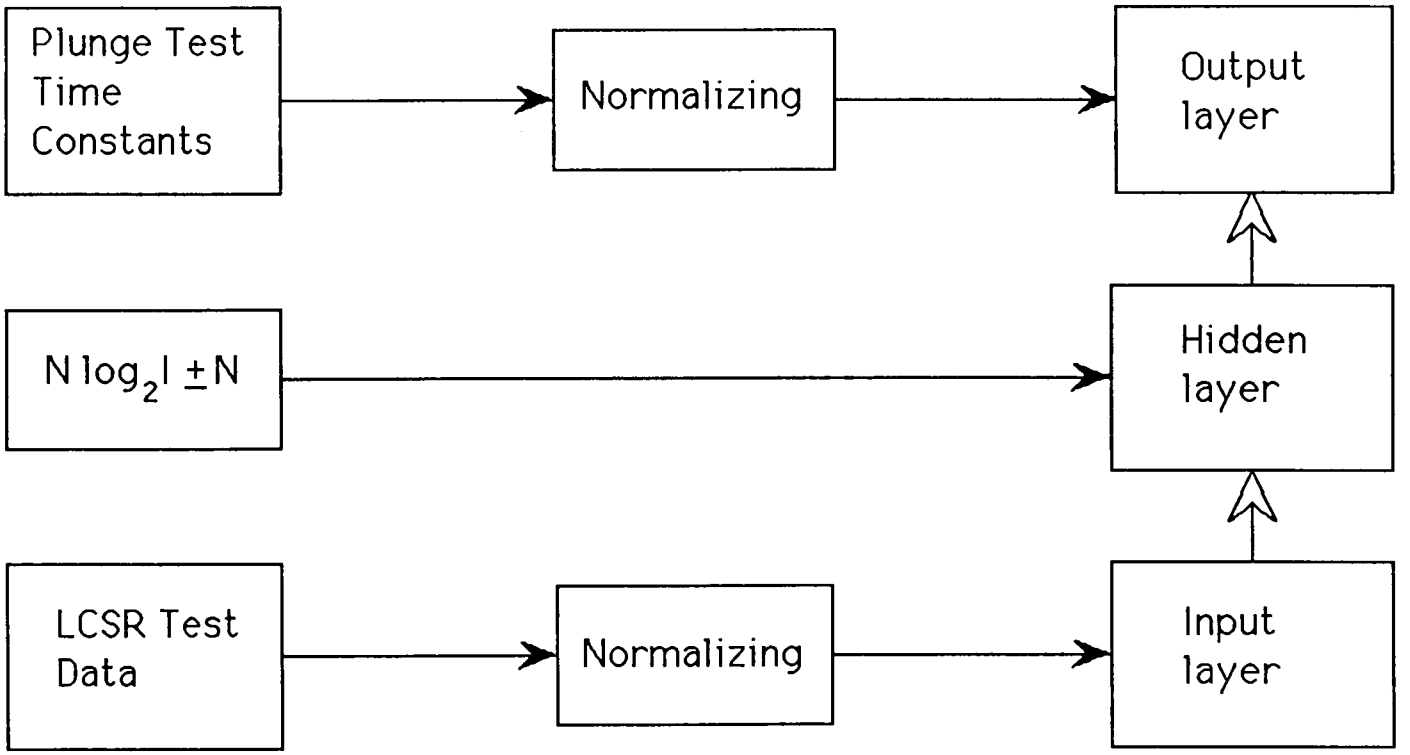



Figure 7.1. Neural Network Training Procedure For Measurement Of Temperature Sensor Response Time

Table 7.1. Neural Network Specifications

| Network | Input-layer PEs | Hidden-layer PEs | Output-layer PEs | Training Cycles | Training Time (hr)* |
|---|---|---|---|---|---|
| Network A | 20 | 41 | 1 | 3000 | 4 |
| Network B | 20 | 65 | 1 | 530 | 1.1 |
| Network C | 20 | 70 | 1 | 6300 | 14 |
| Network D | 40 | 81 | 1 | 2400 | 10 |
| Network E | 40 | 90 | 1 | 2900 | 14.6 |
| Network F | 40 | 100 | 1 | 8200 | 45.6 |
| Network G | 40 | 110 | 1 | 2600 | 17 |
| Network H | 60 | 121 | 1 | 3000 | 33.3 |
| Network I | 60 | 140 | 1 | 4200 | 50 |
| Network J | 60 | 150 | 1 | 2400 | 32 |
| Network K | 60 | 160 | 1 | 2700 | 45 |

* Using computer model Z386/16

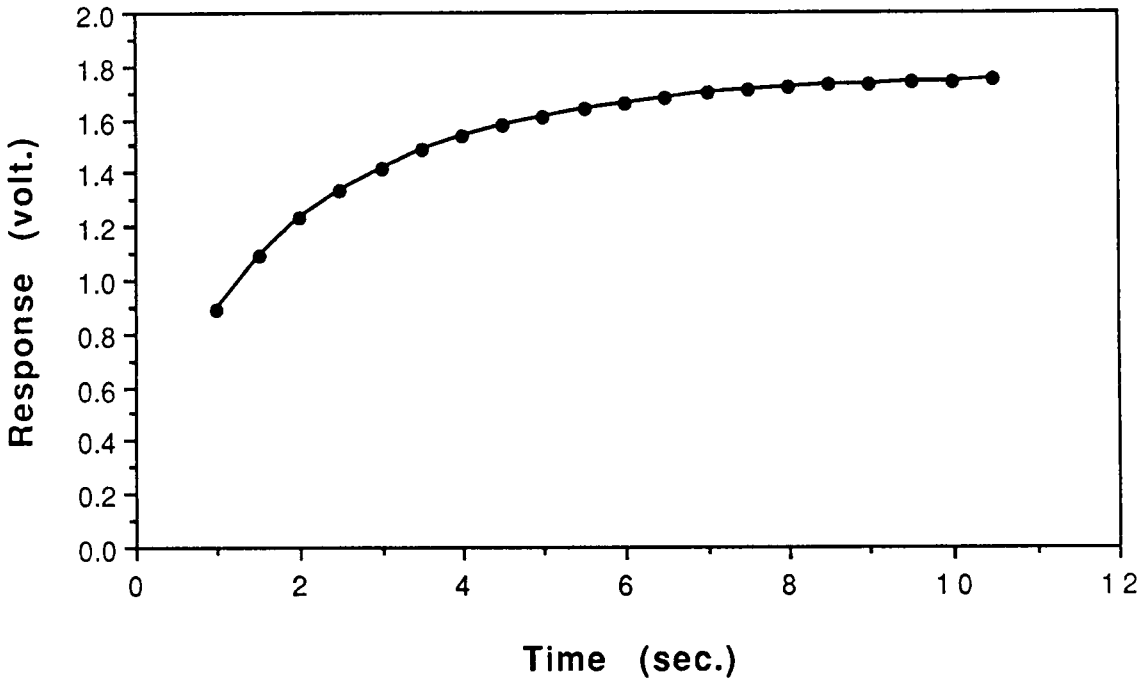



Figure 7.2. An Example Of 20 Input Data For Training Neural Networks

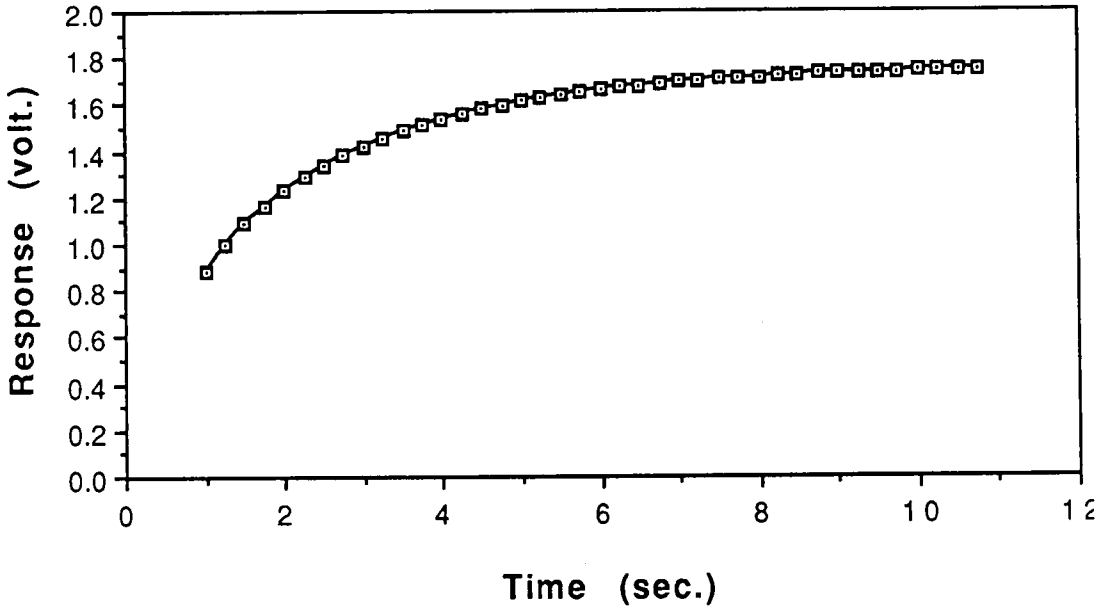



Figure 7.3. An Example Of 40 Input Data For Training Neural Networks

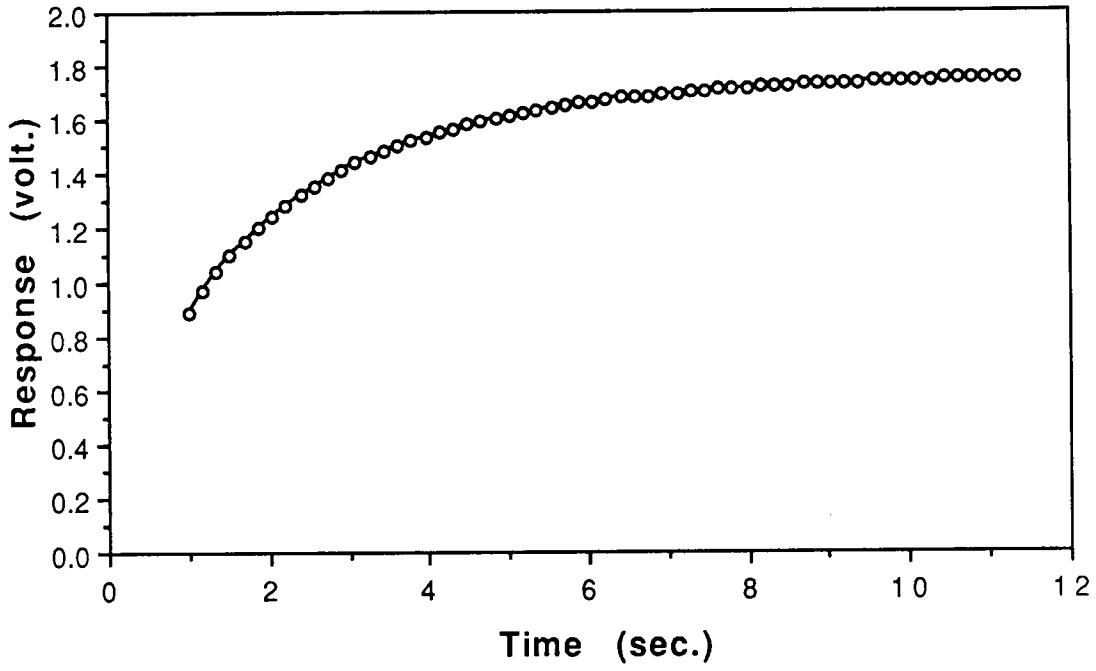



Figure 7.4. An Example Of 60 Input Data For Training Neural Networks

The number of hidden-layer PEs in each network are approximated by using Equation (5.2) and Equation (5.3) [22]. The training is completed when the network-output error has converged to an acceptable error or when the network-output error cannot decrease anymore.

## ❒ RECALL NEURAL NETWORKS

Sets of input/output data used to train a network are often called the training sets of data. Other sets of data not used to train a network may be defined as test sets of data or new sets of data. In this thesis, the training sets of input/output data are obtained from LCSR and plunge tests on RTD A and the test sets of data are obtained from LCSR and plunge tests on RTD B.

A trained network can be recalled to analyze the training sets of data and test sets of data. The analysis of the training sets of data is to verify the network's output response. The analysis of new sets of data is to validate the prediction ability of the network.

### ❖ Recall Neural Networks Using Training Sets Of Data

The trained networks A, B, C, D, E, F, G, H, I, J, and K have been recalled to analyze the training sets of data. The average relative errors of the analysis are shown in Figure 7.5. The highest average relative error is 4.61 percent obtained by Network A, consisting of

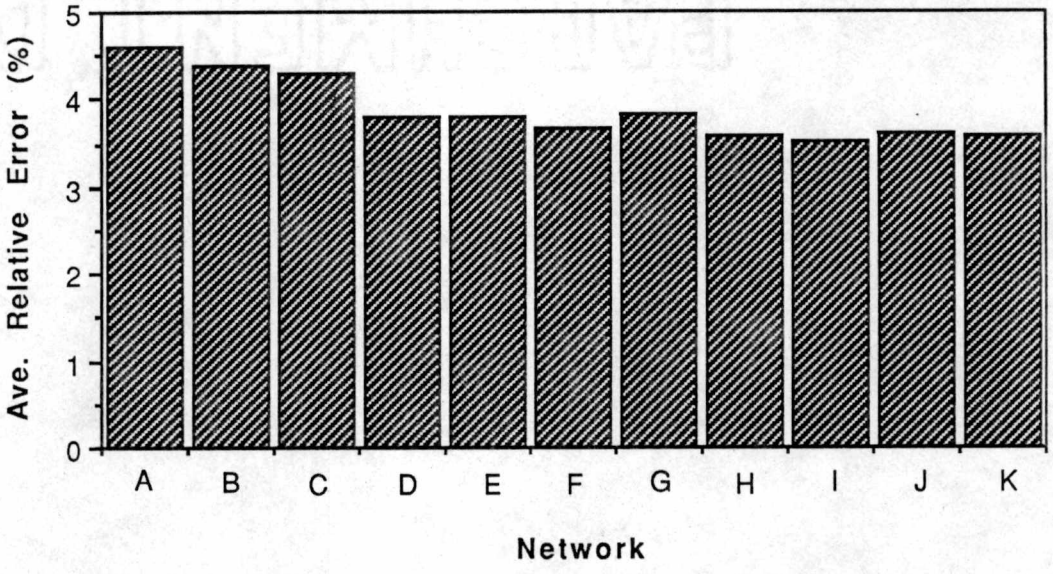



Figure 7.5. The Average Relative Error Of The Time Constant Predictions Of The Networks Using Training Sets Of Data

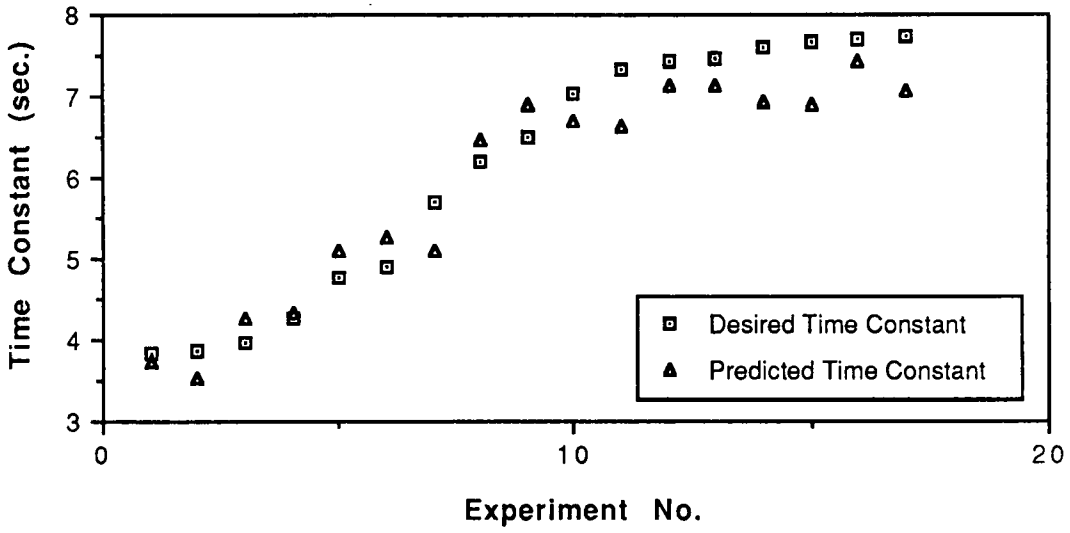
percent. However, its training time is much longer than Network A's, because Network I's architecture consisting of 60 input-layer PEs, 140 hidden-layer PEs, and 1 output layer PE is more complicated than Network A's.

### ❖ Recall Neural Networks Using Test Sets Of Data

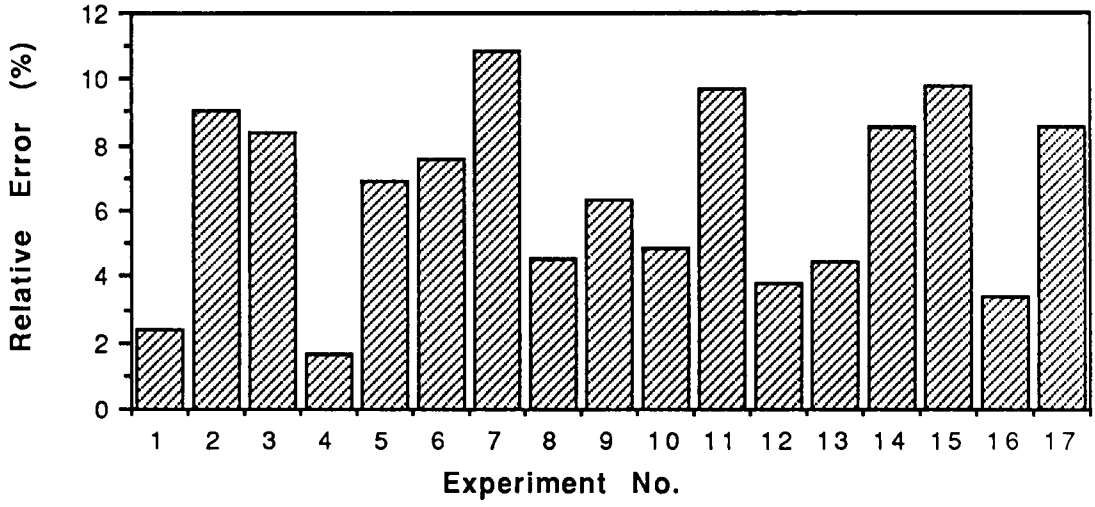
Test sets of data obtained from RTD B have been processed by those trained networks listed in Table 7.1 to validate their abilities to predict the time constants from the LCSR transients. The results of the validations are shown in Figure 7.6 to Figure 7.16, respectively. Figure 7.17 shows the average relative error of the time constant predictions of the networks.

As shown in Table 7.1, networks A, B, and C use 20 input-layer PEs and different number of hidden-layer PEs. Network A employs 41 hidden-layer PEs, Network B employs 65 hidden-layer PEs, and Network C uses 70 hidden-layer PEs. The variation in the number of hidden-layer PEs yields different time-constant predictions of those networks. The average relative error of Network A's predictions is 6.48 percent, the average relative error of Network B's predictions is 6.34 percent, and the average relative error of Network C's predictions is 6.50 percent.

These errors are reduced by increasing the number of input-layer PEs of the networks. The networks that employ 40 input-layer PEs are networks D, E, F, and G. The number of hidden-layer PEs of these networks are approximated by Equation (5.2) and Equation (5.3) [22]. The average errors of networks D, E, F, and G are 5.44 percent,

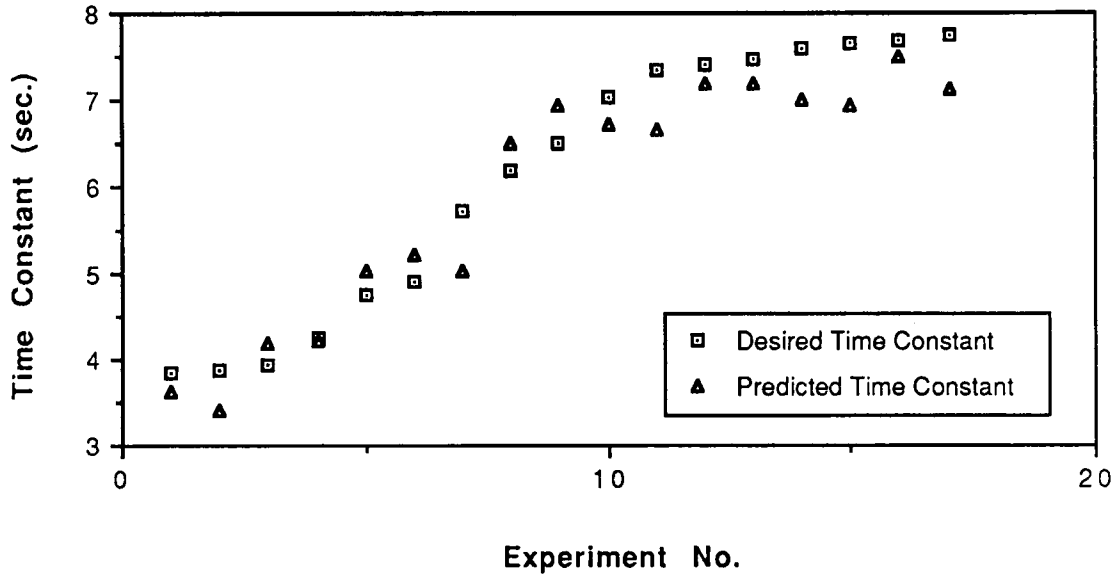



(a) Time Constant Comparison

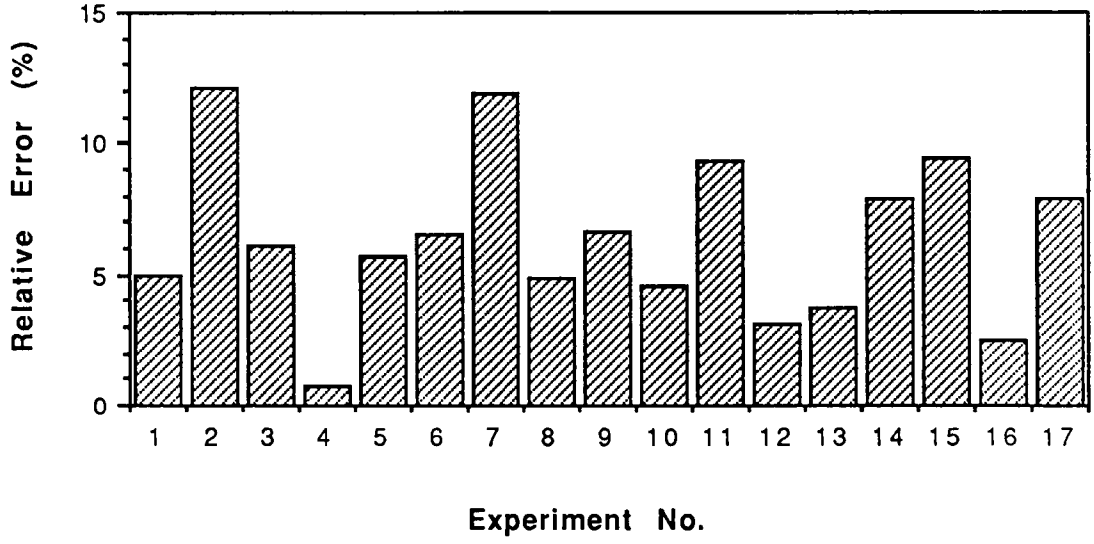



(b) Relative Error Of The Predicted Time Constant

Figure 7.6. Time Constant Predictions Of Network A Using Test Sets Of Data

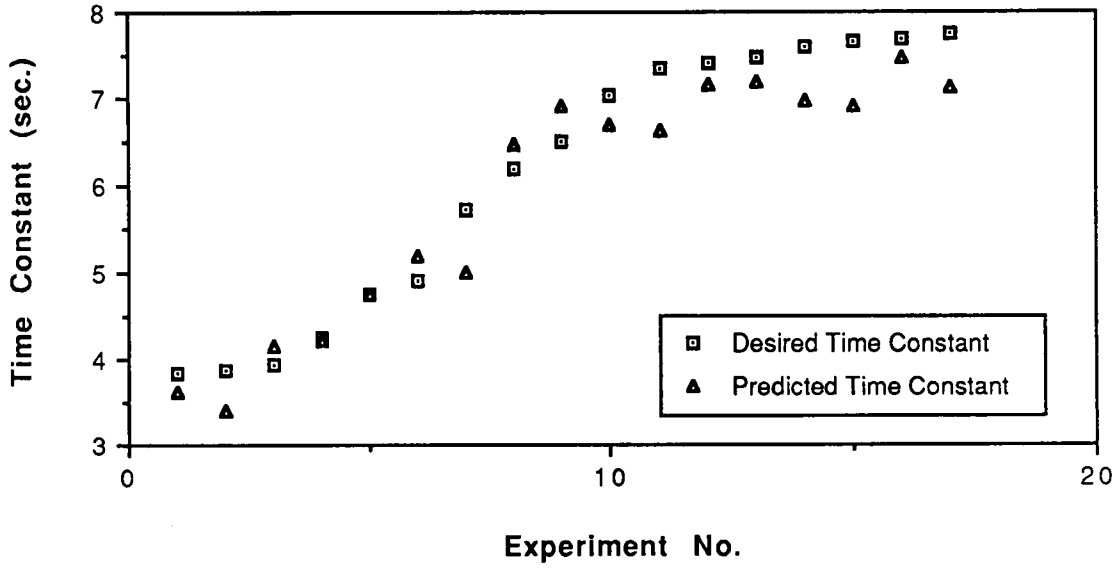



(a) Time Constant Comparison

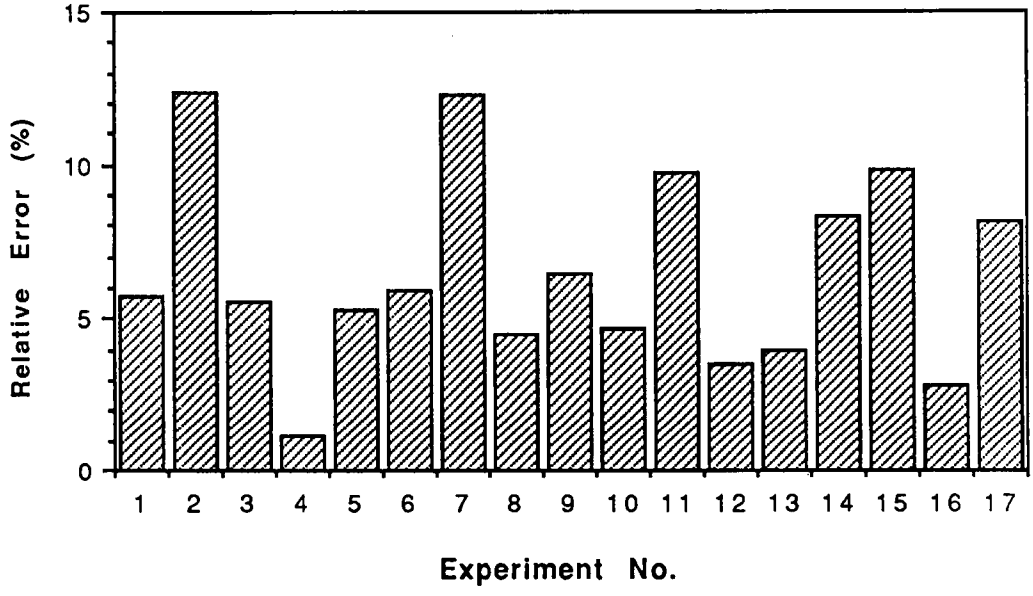



(b) Relative Error Of The Predicted Time Constant

Figure 7.7. Time Constant Predictions Of Network B Using Test Sets Of Data

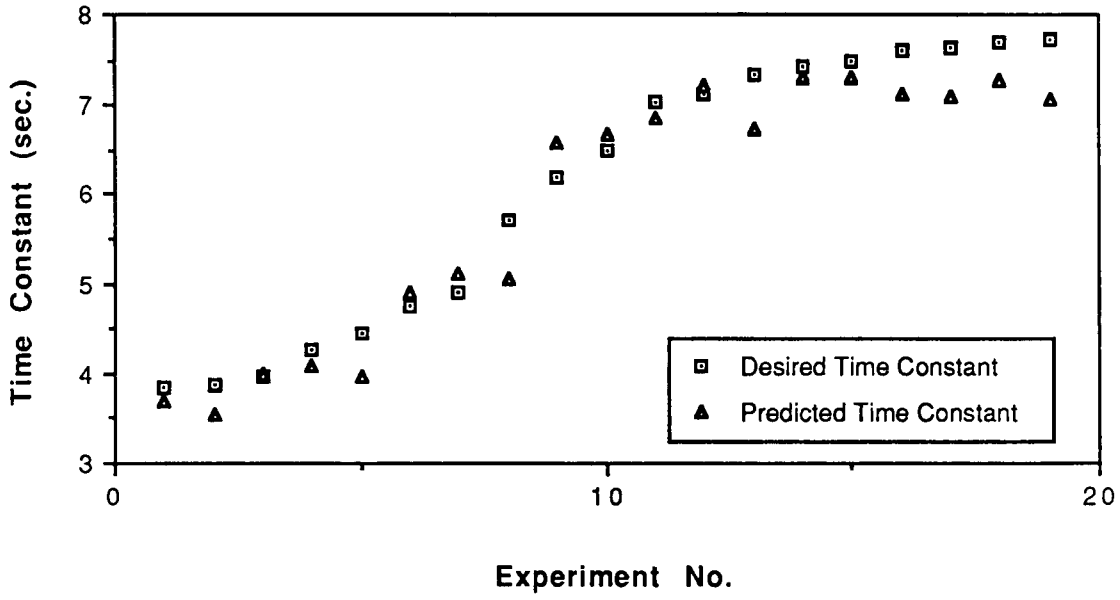



(a) Time Constant Comparison




(b) Relative Error Of The Predicted Time Constant

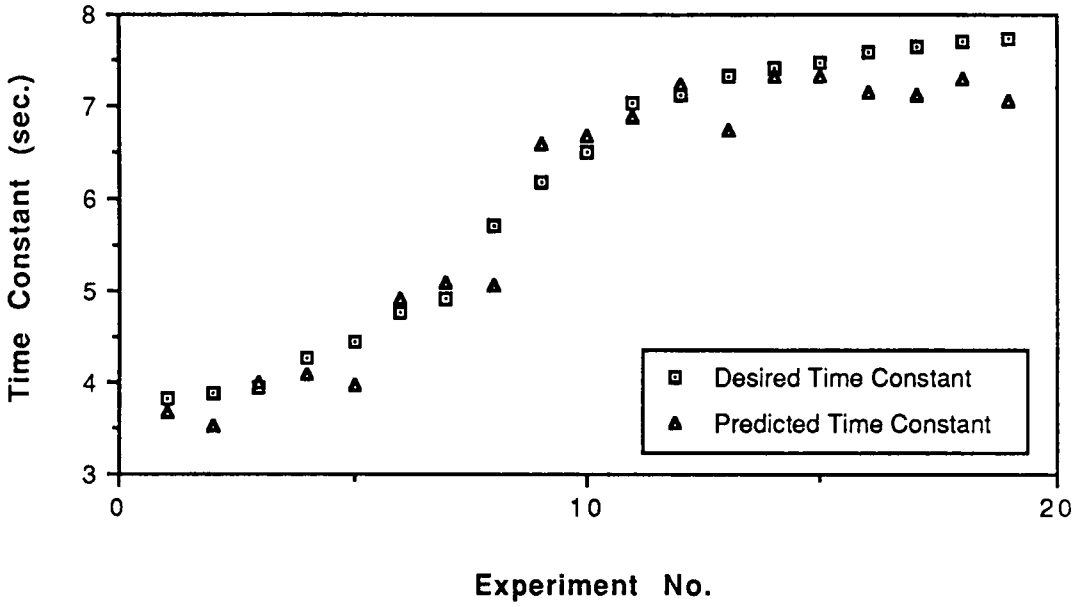
Figure 7.8. Time Constant Predictions Of Network C Using Test Sets Of Data



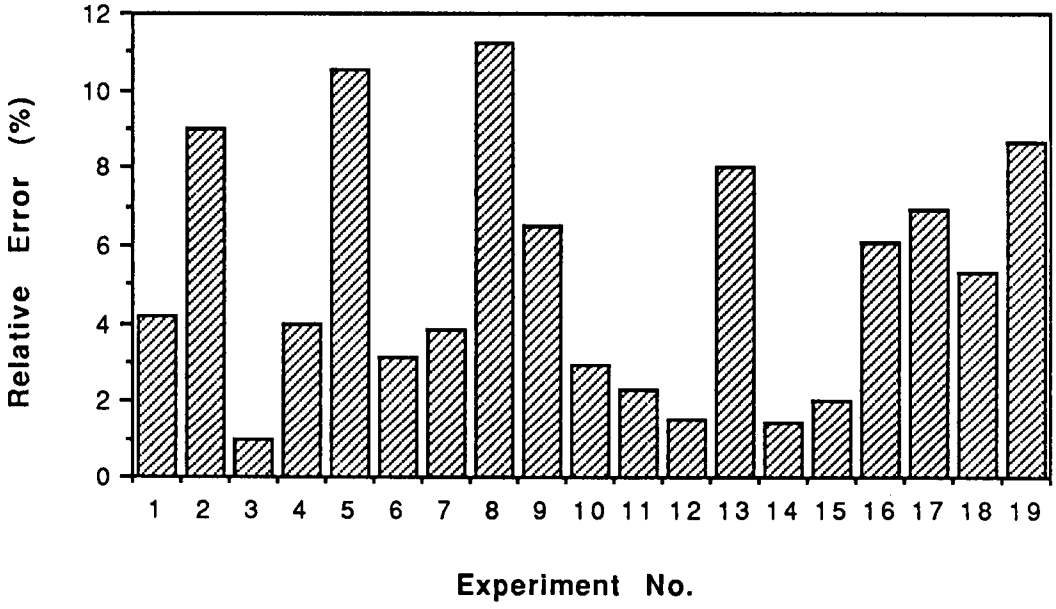

(a) Time Constant Comparison


(b) Relative Error Of The Predicted Time Constant

Figure 7.9. Time Constant Predictions Of Network D Using Test Sets Of Data

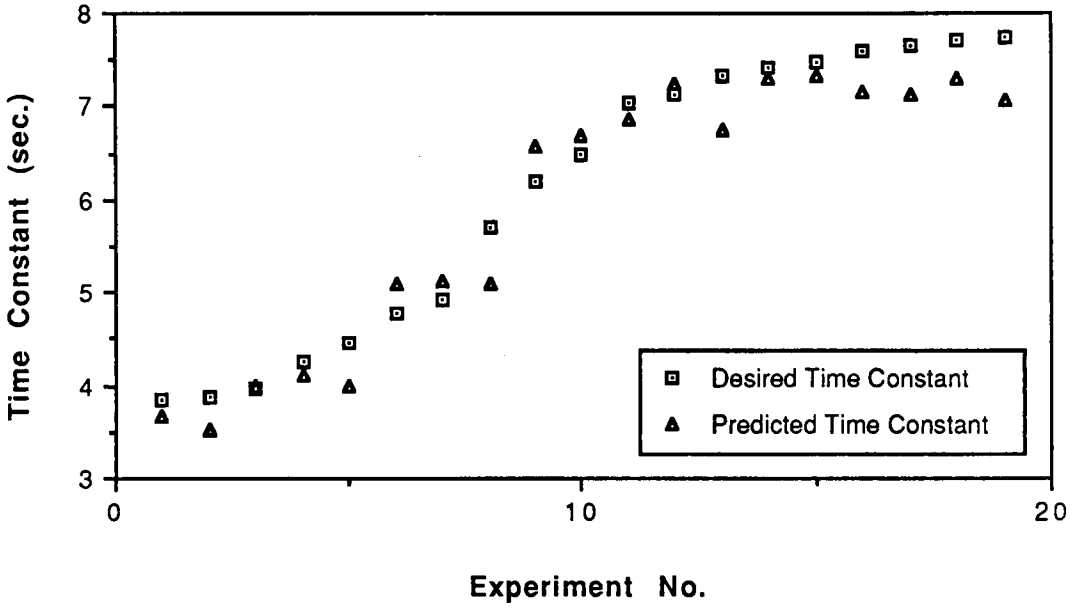



(a) Time Constant Comparison

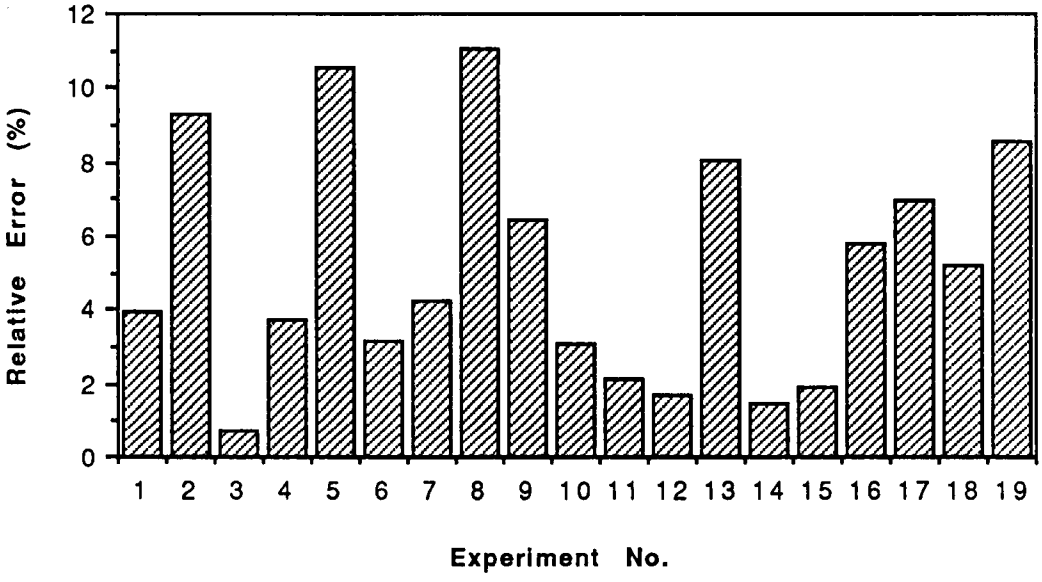



(b) Relative Error Of The Predicted Time Constant

Figure 7.10. Time Constant Predictions Of Network E Using Test Sets Of Data

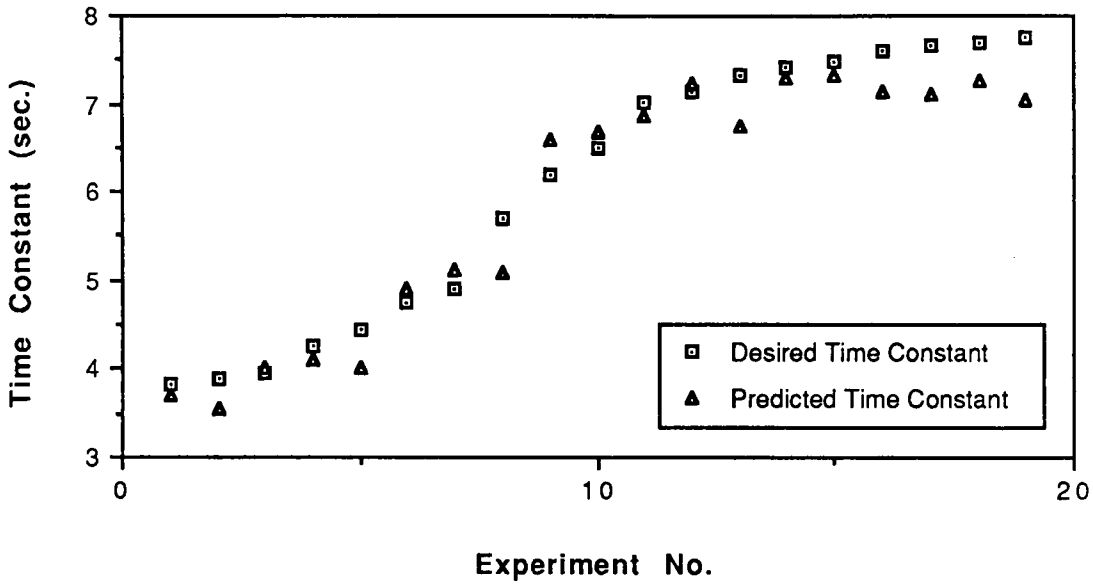



(a) Time Constant Comparison

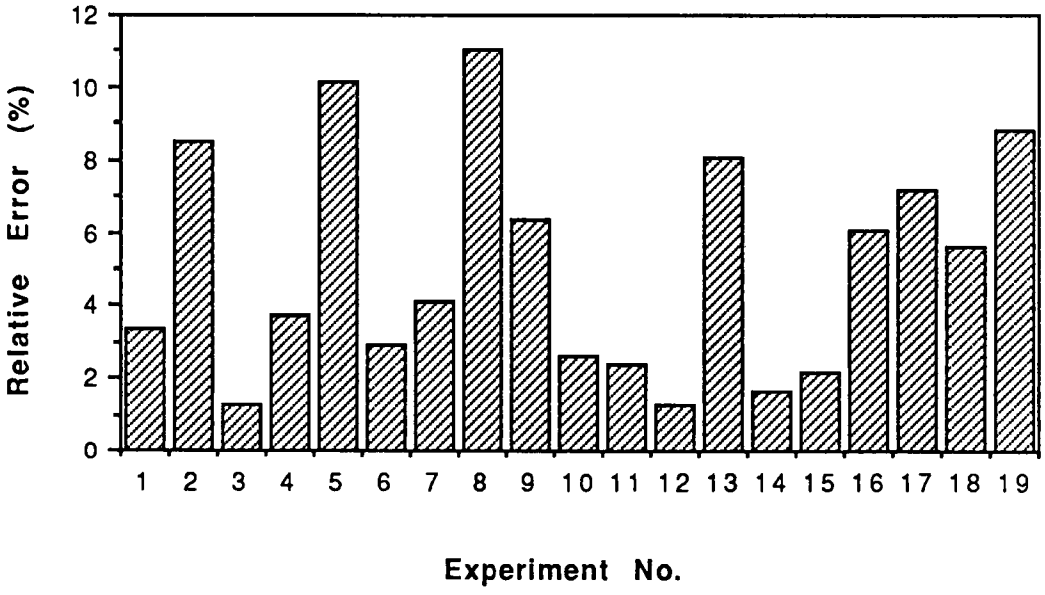



(b) Relative Error Of The Predicted Time Constant

Figure 7.11. Time Constant Predictions Of Network F Using Test Sets Of Data

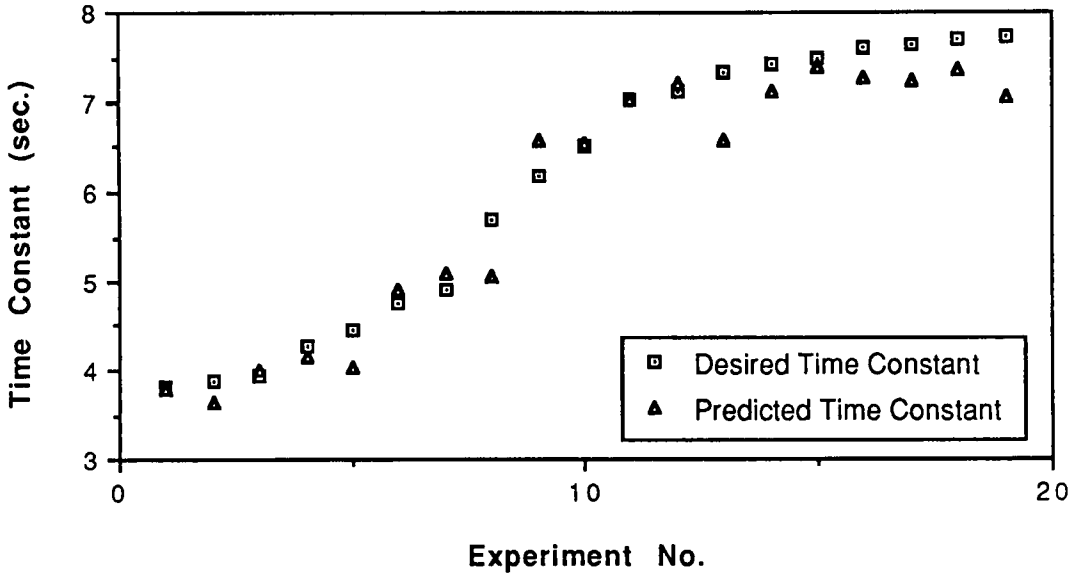



(a) Time Constant Comparison

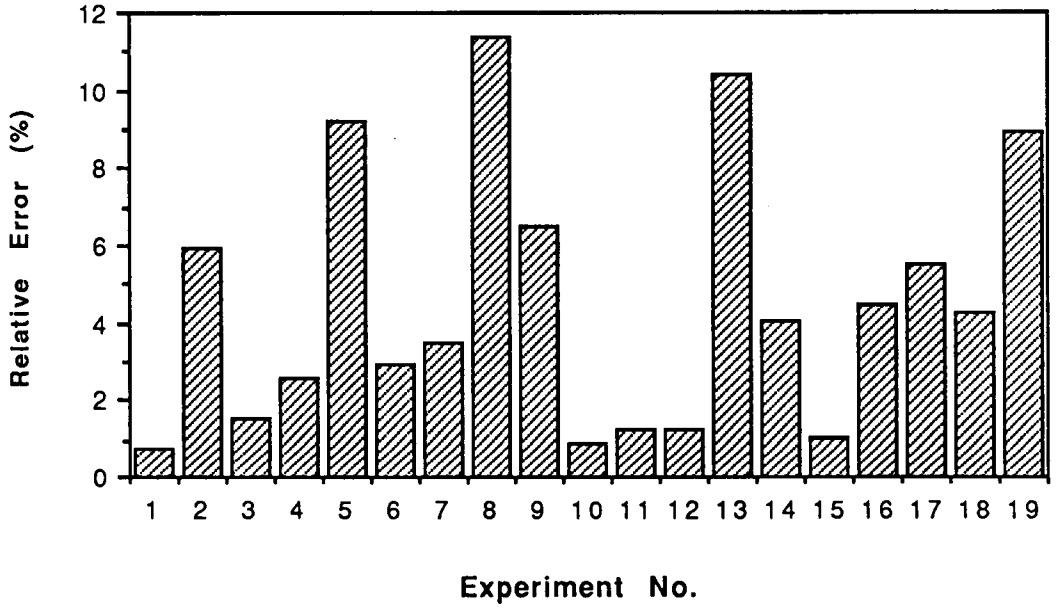



(b) Relative Error Of The Predicted Time Constant

Figure 7.12. Time Constant Predictions Of Network G Using Test Sets Of Data

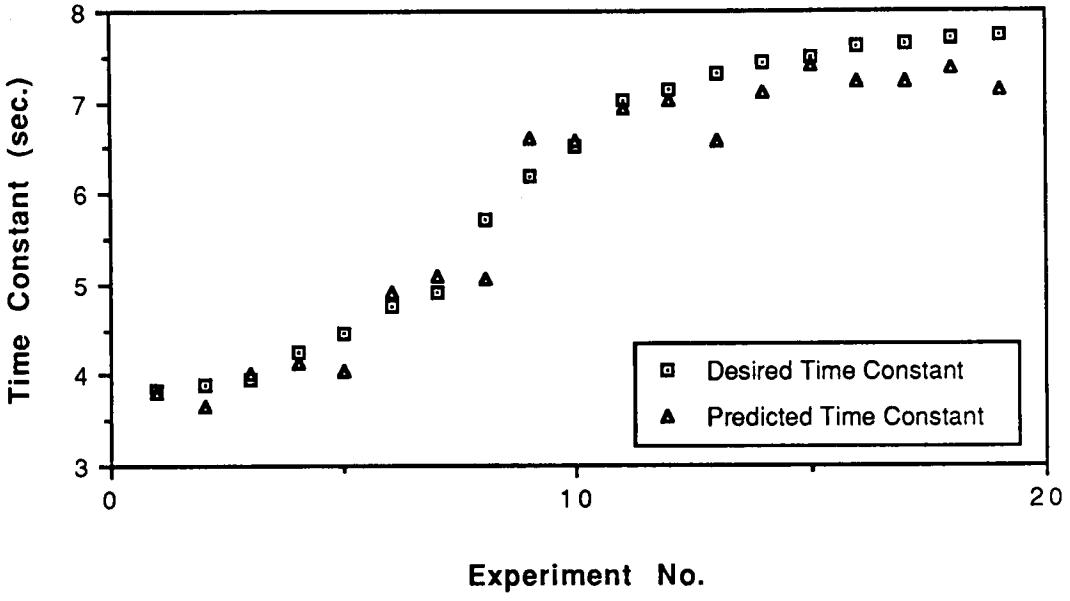



(a) Time Constant Comparison

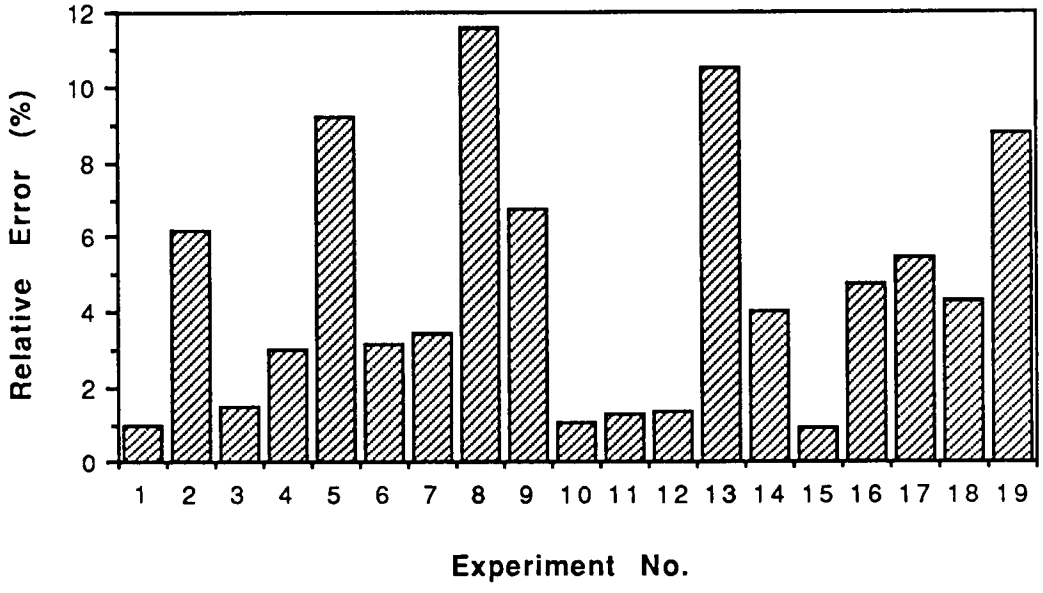



(b) Relative Error Of The Predicted Time Constant

Figure 7.13. Time Constant Predictions Of Network H Using Test Sets Of Data

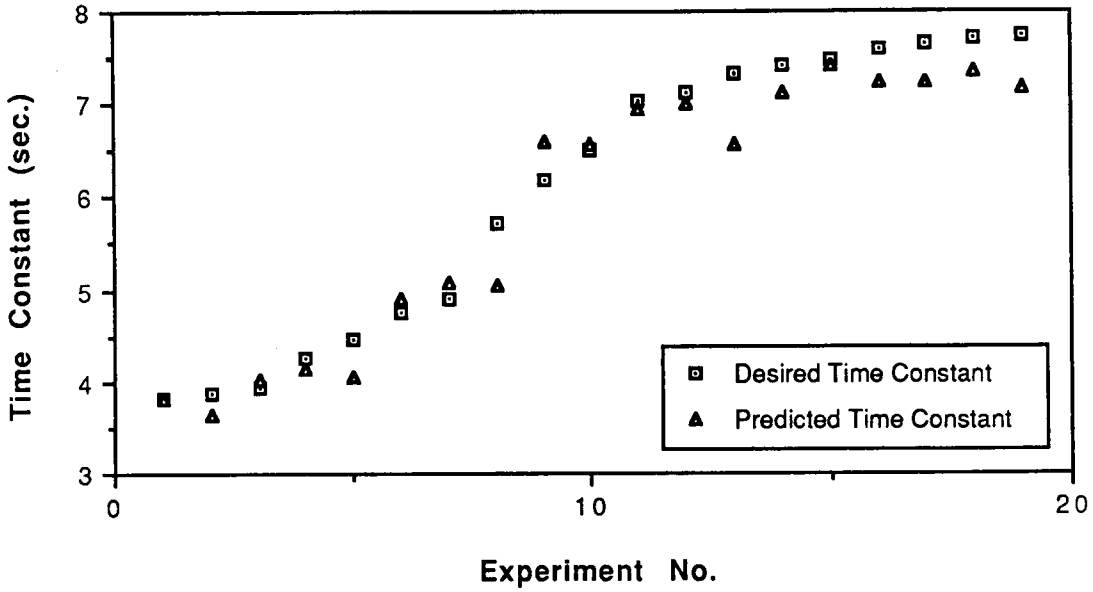



(a) Time Constant Comparison

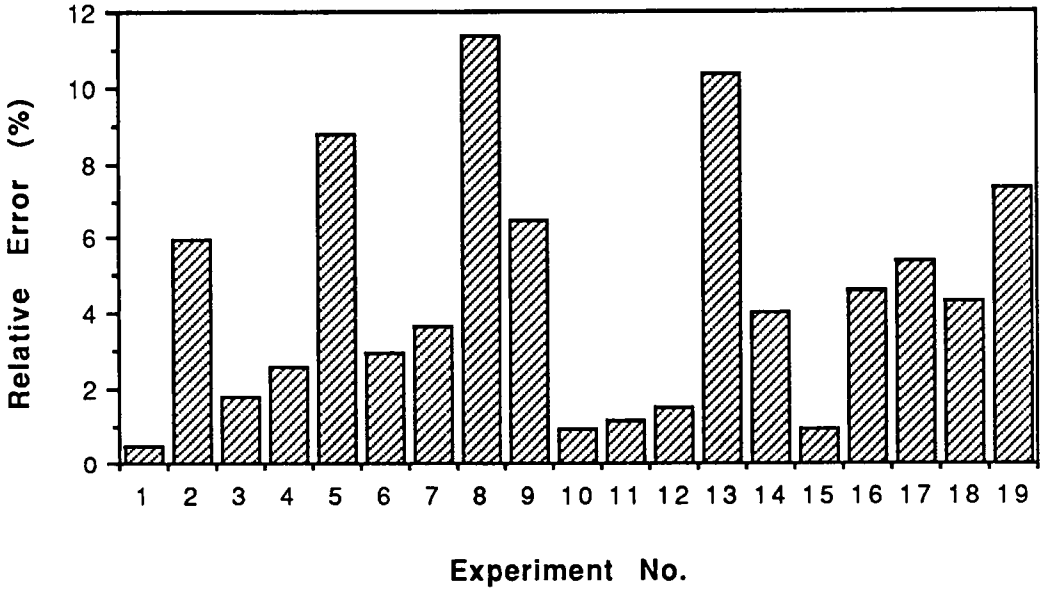



(b) Relative Error Of The Predicted Time Constant

Figure 7.14. Time Constant Predictions Of Network I Using Test Sets Of Data

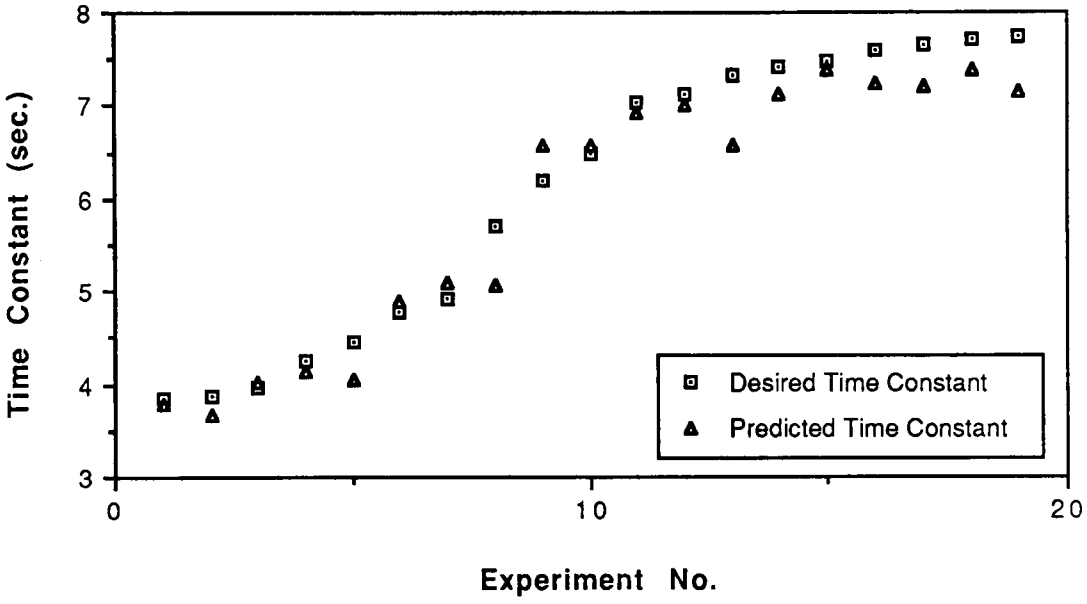



(a) Time Constant Comparison

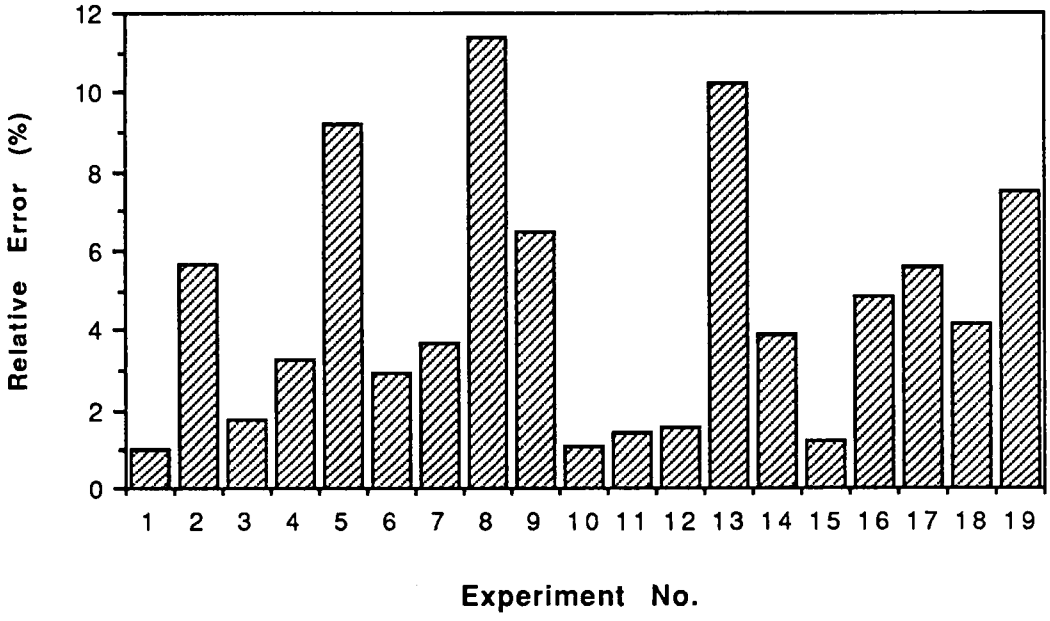



(b) Relative Error Of The Predicted Time Constant

Figure 7.15. Time Constant Predictions Of Network J Using Test Sets Of Data




(a) Time Constant Comparison




(b) Relative Error Of The Predicted Time Constant

Figure 7.16. Time Constant Predictions Of Network K Using Test Sets Of Data

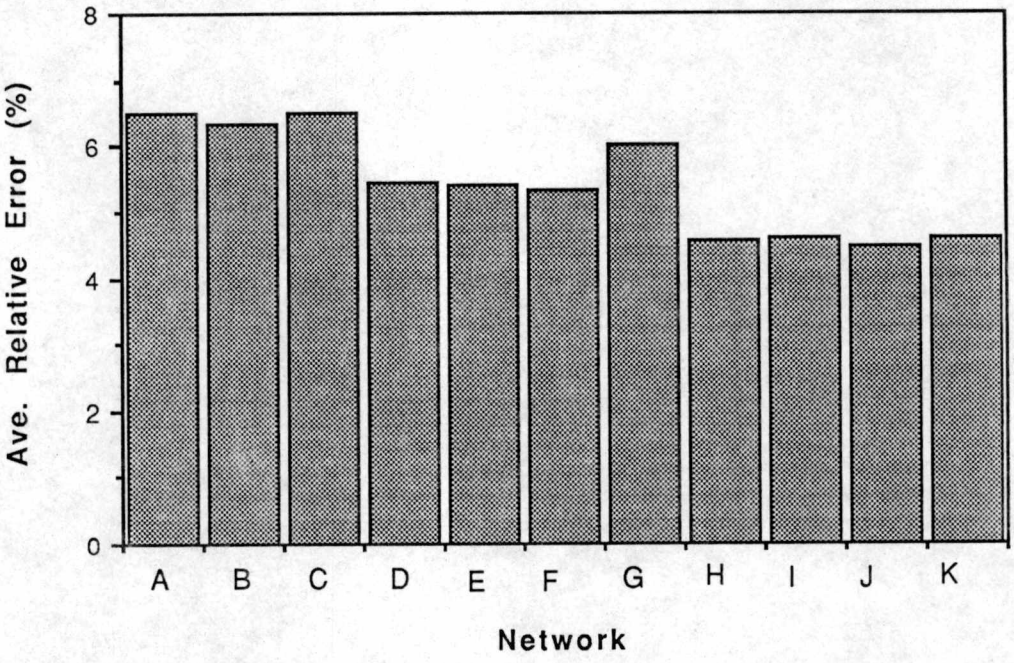


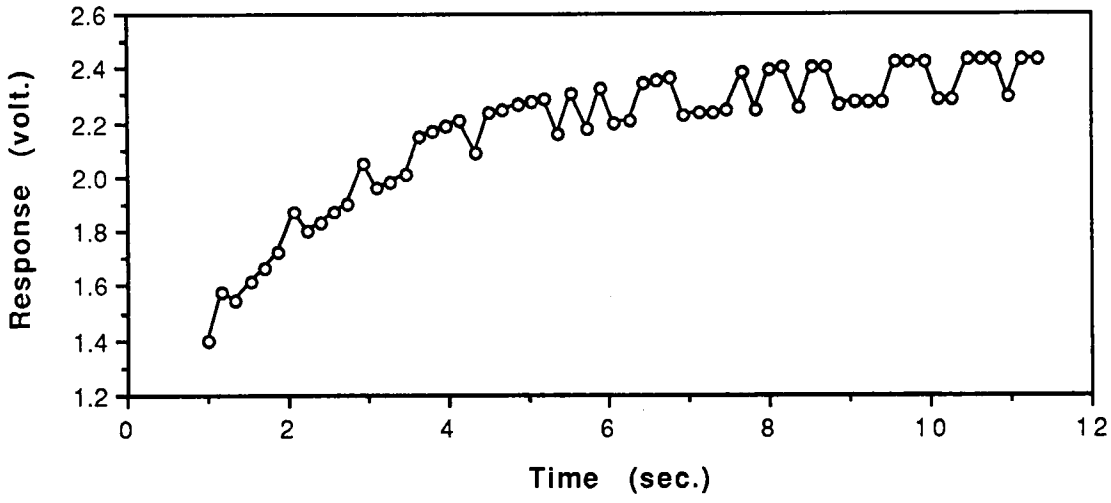

Figure 7.17. The Average Relative Error Of The Time Constant Predictions Of The Networks Using Test Sets Of Data

5.39 percent, 5.33 percent, and 6.03 percent, respectively.

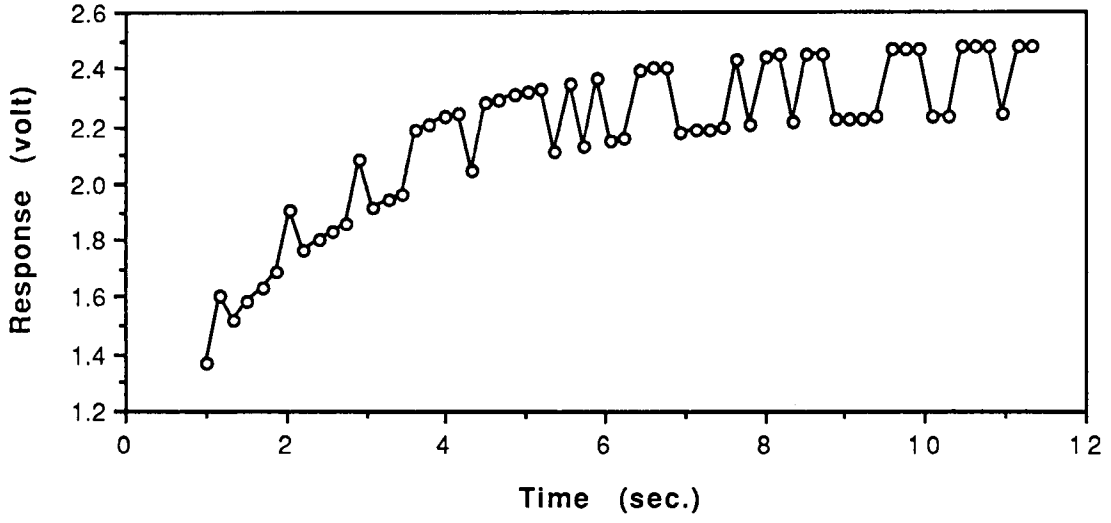
The last four networks, which are networks H, I, J, and K, use 60 input-layer PEs to minimize the errors of the networks' time constant predictions. The number of hidden-layer PEs of networks H, I, J, and K is estimated by Equation (5.2) and Equation (5.3) [22]. The networks reduce the average relative error to within 5 percent. However, the training times of networks H, I, J, and K are much longer than those of the previous networks.

## ❐ PREDICTIONS OF NEURAL NETWORKS USING PERTURBED SETS OF DATA

Sometimes the LCSR response transients obtained from a nuclear power plant are contaminated by noise. The noise generates inaccuracies in measuring the time constants of the sensors. In order to anticipate this problem, 3 and 5 percent simulated noise have been added to the training sets of input/output data as shown in Figure 7.18. Two networks, consisting of 60 input-layer PEs, 150 hidden-layer PEs, and 1 output-layer PE, have been trained with those training sets of data. The trained networks have been recalled to analyze test sets of data contaminated by 3 and 5 percent noises. The results of the analysis using training sets of contaminated data are shown in Figure 7.19 and 7.20. The average relative errors of the time constant prediction using the training sets of data are 3.42 percent for 3 percent noise and 3.52 percent for 5 percent noise.

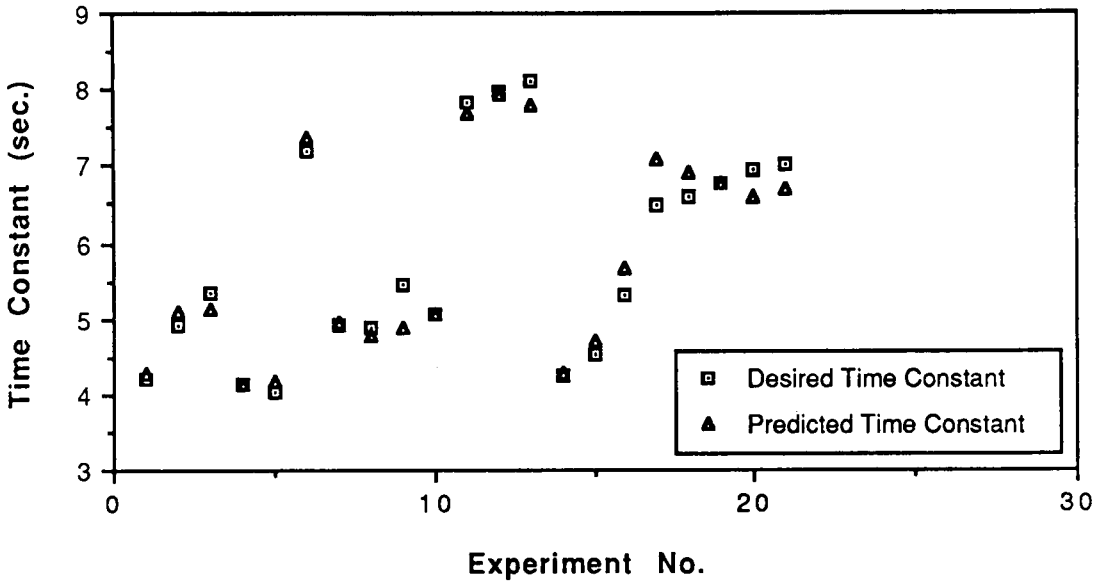



(a) Three Percent Noise

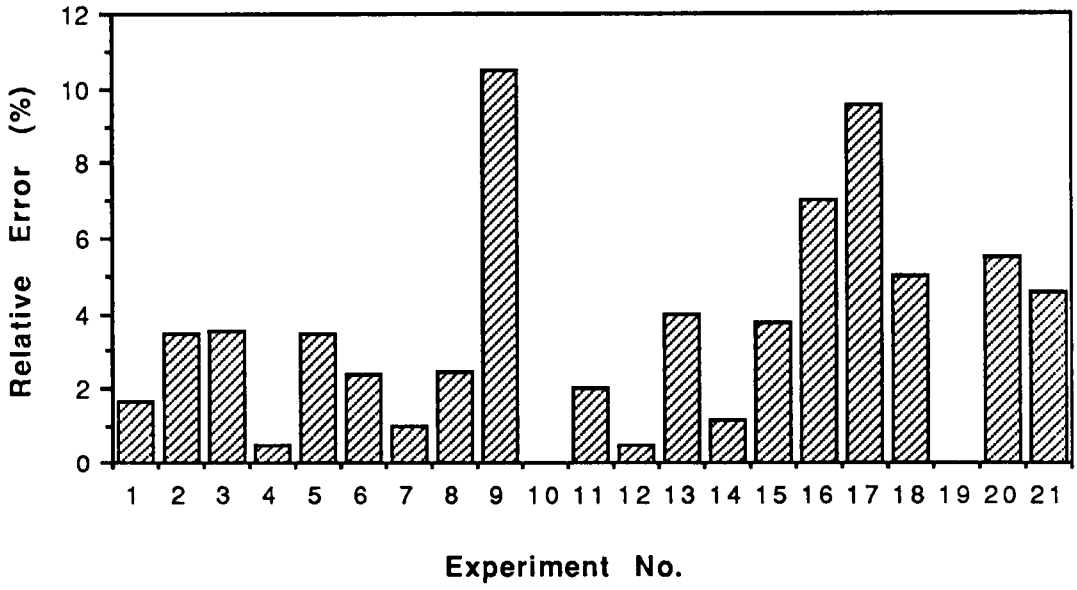



(b) Five Percent Noise

Figure 7.18. An Example Of 60 Input Data Contaminated With Noise For Training The Network

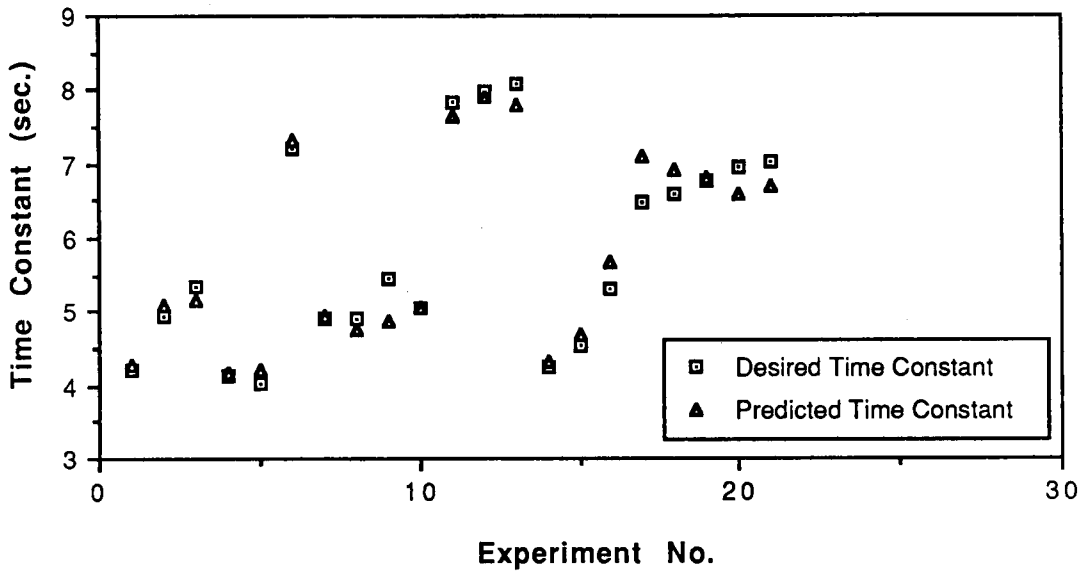



(a) Time Constant Comparison

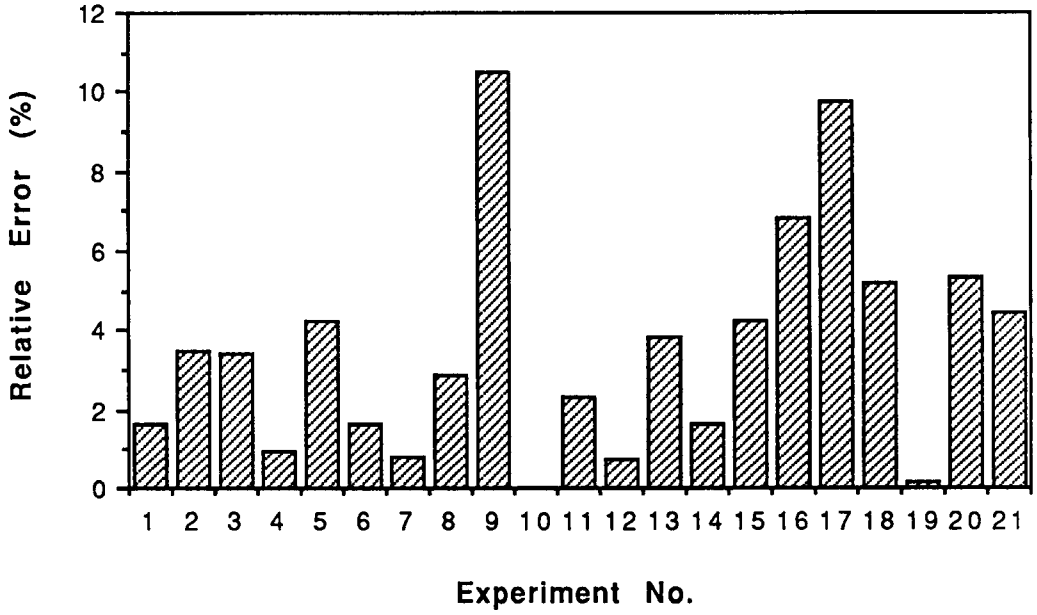



(b) Relative Error Of The Predicted Time Constant

Figure 7.19. Time Constant Predictions Of The Network Using Training Sets Of Data Contaminated With 3 Percent Noise




(a) Time Constant Comparison



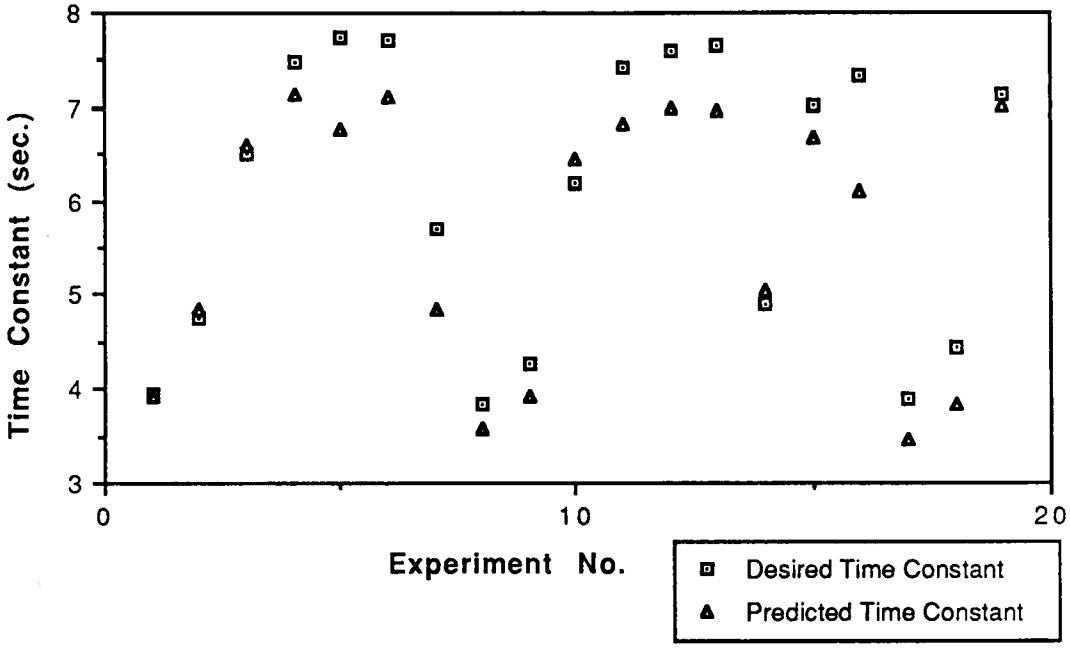

(b) Relative Error Of The Predicted Time Constant

Figure 7.20. Time Constant Predictions Of The Network Using Training Sets Of Data Contaminated With 5 Percent Noise

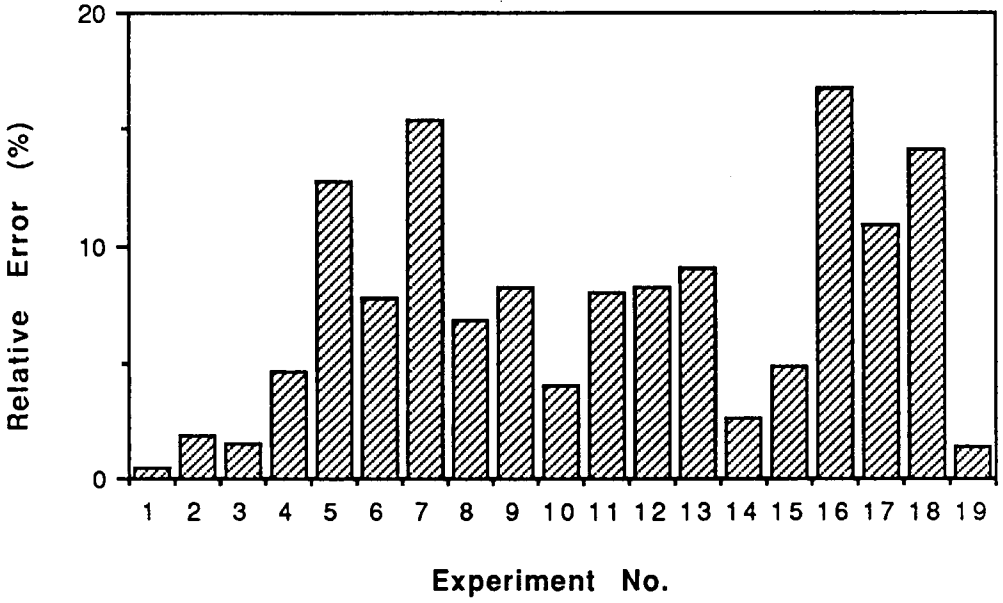
Figure 7.21 and 7.22 show the results of the time constant predictions of the networks using test sets of contaminated data. The average relative errors of the time constant predictions are 7.33 percent for 3 percent noise and 7.96 for 5 percent noise. Those errors indicate that the networks are still able to predict satisfactorily the sensor's time constant from LCSR data contaminated with noise.

## ❒ PREDICTIONS OF NEURAL NETWORKS USING SETS OF DATA OBTAINED FROM IMPERFECT EQUIPMENT

As mentioned in Chapter 6, in order to identify the network's ability to handle equipment imperfections, low-wattage resistors have been used in Wheatstone bridge circuit to simulate an imperfect equipment. In LCSR tests, the resistances of the low-wattage resistors will change due to an increased current through the bridge. The imperfect equipment may produce imprecise sets of input/output data. These sets of input/output data are used to train a network, which consists of 60 input-layer PEs and 150 hidden-layer PEs, 6000 training cycles. The trained network is then used to analyze the training sets of data and test sets of data. The results of the recall are shown in Figure 7.23 and 7.24. The average relative error using the training sets of input/output data is 7.12 percent. The smallest error is 0.55 percent and the highest error is 16.12 percent. The results of the recall using test sets on data are better than those using the training sets of input/output data. The average relative error using test sets of input data is 6.02 percent. The

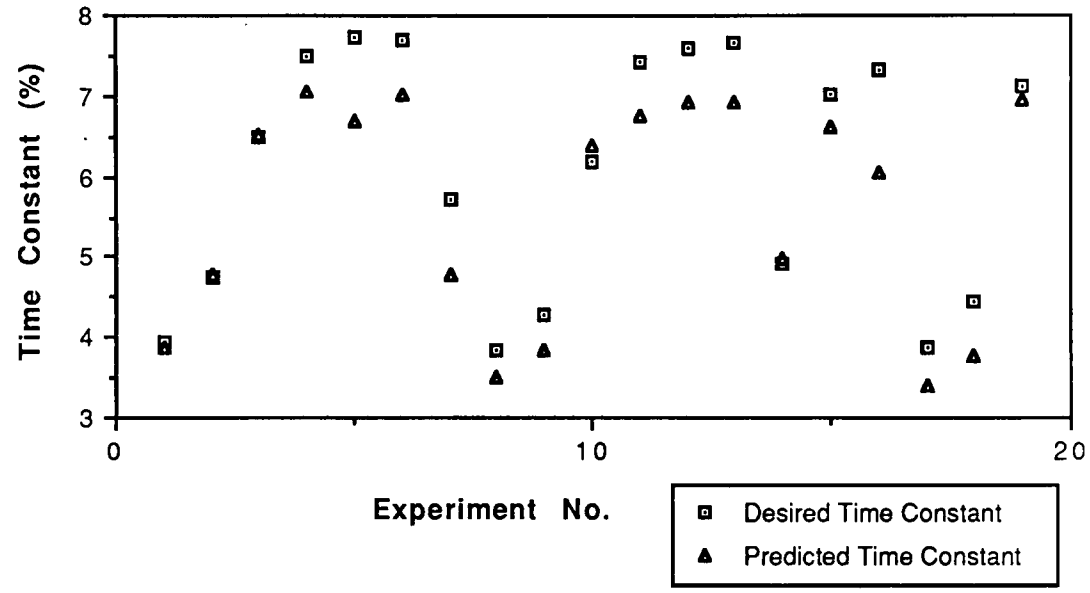



(a) Time Constant Comparison

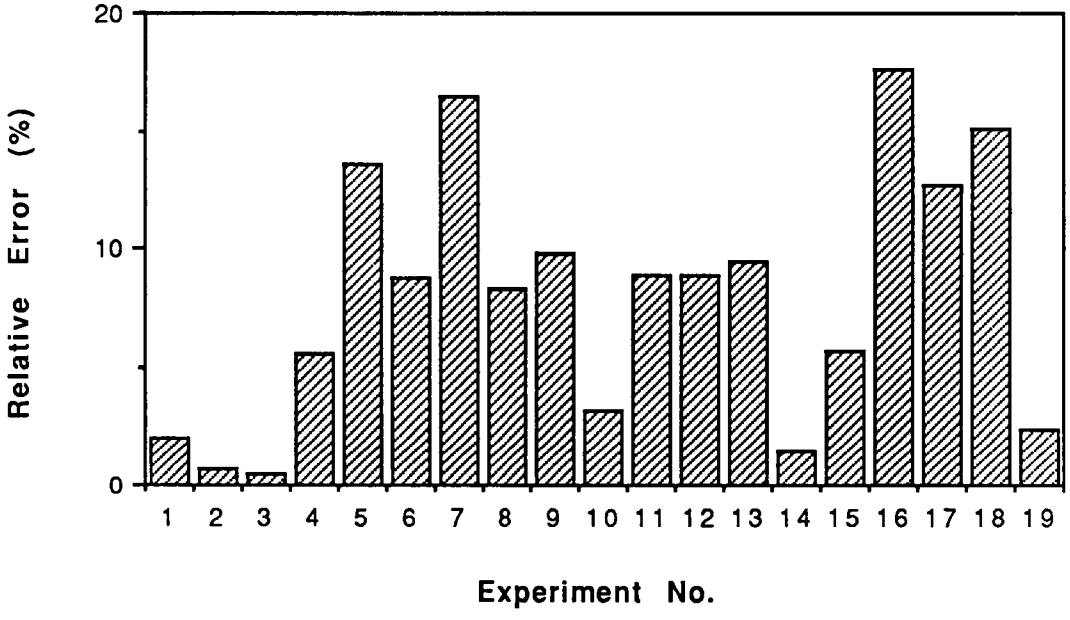



(b) Relative Error Of The Predicted Time Constant

Figure 7.21. Time Constant Predictions Of The Network Using Test Sets Of Data Contaminated With 3 Percent Noise

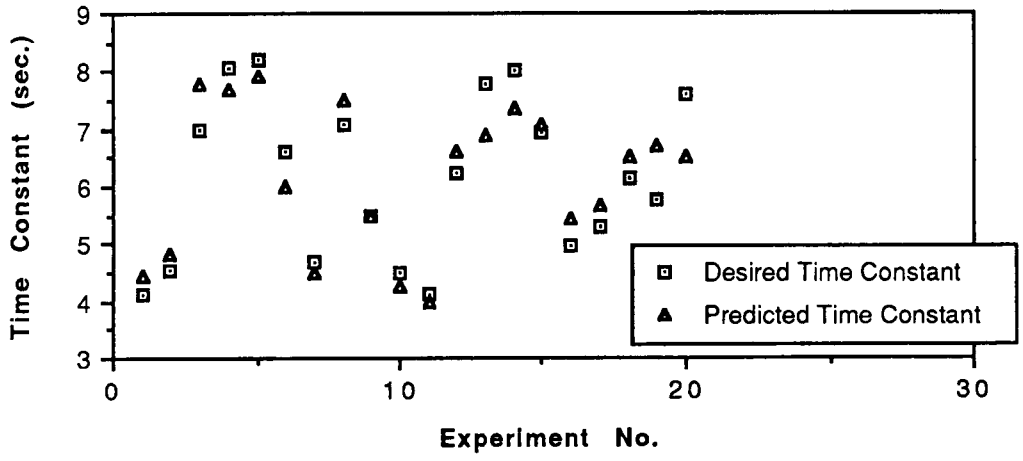



(a) Time Constant Comparison

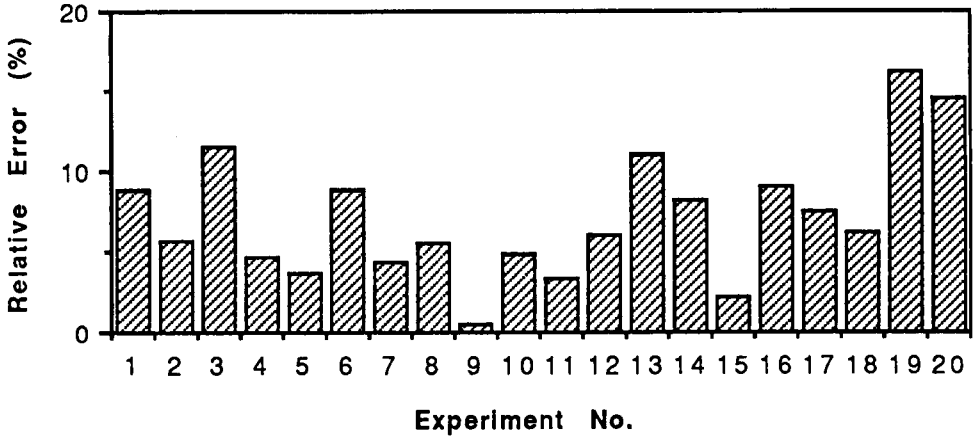



(b) Relative Error Of The Predicted Time Constant

Figure 7.22. Time Constant Predictions Of The Network Using Test Sets Of Data Contaminated With 5 Percent Noise

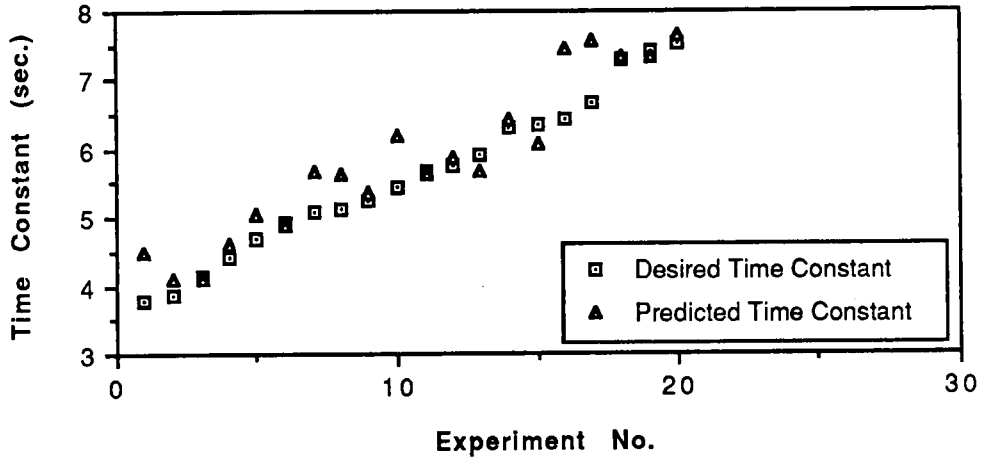



(a) Time Constant Comparison

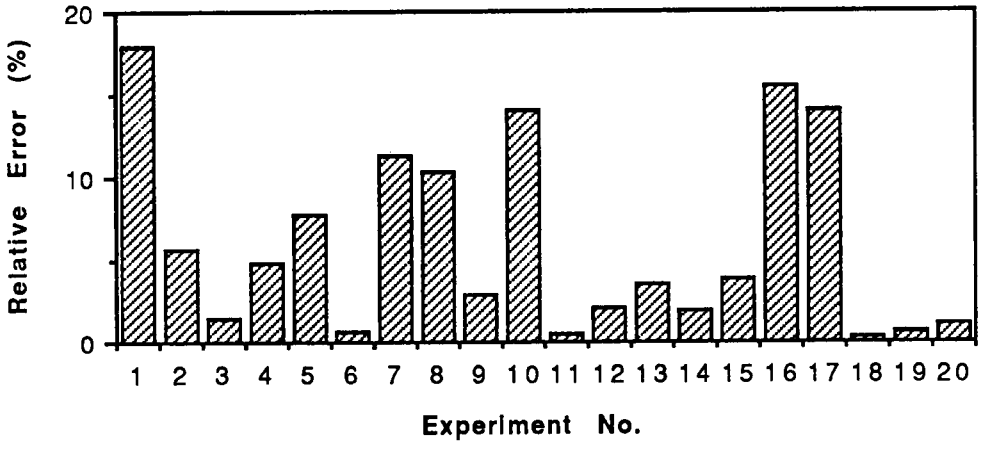



(b) Relative Error Of The Predicted Time Constant

Figure 7.23. Time Constant Predictions Of The Network Using Training Sets Of Input/OutputData Obtained From Imperfect Equipment




(a) Time Constant Comparison




(b) Relative Error Of The Predicted Time Constant

Figure 7.24. Time Constant Predictions Of The Network Using Test Sets Of Input/OutputData Obtained From Imperfect Equipment

smallest error is 0.53 percent and the highest error is 15.53 percent. These results indicate that backpropagation networks are able to overcome imperfections in the test equipment.

# CHAPTER 8
# CONCLUSIONS

The backpropagation neural network is able to predict the time constant of the RTD from LCSR test data. The time constant predictions of the trained networks both using the training sets of input/output data and using the test sets of input data produce the average relative error of about 7 percent. These results indicate that the time constant prediction of the backpropagation network from LCSR test transients is better than that of the LCSR analytical transformation involving complicated computation, highly trained personnel, and specialized equipment. Another advantage of the neural network analysis is that it can be routinely applied at very little cost.

Neural networks have been trained using sets of input/output data. The time constants obtained from the plunge tests are used as desired output. The input data are selected from the LCSR response transients at equal intervals between 1 and 12 seconds. The training of the networks is complete when the network-output errors have converged to an acceptable error. The lowest average relative error is 4.45 percent obtained by Network J, which consists 60 input-layer PEs, 150 hidden-layer PEs, and 1 output-layer PE.

Two networks, consisting of 60 input-layer PEs, 150 hidden-layer PEs, and 1 output-layer PE, have been trained with training sets of data contaminated with 3 and 5 percent noise. The trained networks have been recalled to predict the time constants using test sets of data contaminated with 3 and 5 percent noise. The results of these analysis indicate that the network is able to predict accurate-

ly the sensor time constants in the presence of noise. The average relative error of the predictions is within 8 percent.

A network consisting of 60 input-layer PEs, 150 hidden-layer PEs, and 1 output-layer PE has been trained using 20 sets of input/output data obtained from an imperfect equipment. The results of the recall using new sets of input data are satisfactory. The average relative error is 6.02 percent, which indicates that backpropagation neural networks have the potential to overcome equipment imperfections.

# LIST OF REFERENCES

# LIST OF REFERENCES


[1]. Carrol, R.M., Shepard, R.L., and Kerlin, T.W., "In-situ Measurements of The Response Time of Sheathed Thermocouple," *Trans. American Nuclear Society*, Vol. 21, June 1975

[2]. T. W. Kerlin, Miller, L.F., Mott, J.E., Upadhyaya, B.R., Hashemian, H.M., and Arendt, J.S., "In-Situ Response Time Testing Of Platinum Resistance Thermometers," *Electric Power Research Institute*, EPRI NP-459, January 1977.

[3]. Hashemian, H.M., "In-situ Response Time Testing of Platinum Resistance Thermometers in Nuclear Power Plants," *Thesis*, The University of Tennessee, Knoxville, December 1977

[4]. Poore, W.P., "Resistance Thermometer Characteristics and Response Time Testing," *Thesis*, The University of Tennessee, Knoxville, December 1979

[5]. Kerlin, T.W., Shepard, R.L., Hashemian, H.M., and Petersen, K.M., "Response of Installed Temperature Sensors," *Temperature: Its Measurement and Control in Science and Industry*, American Institute of Physics, Vol. 5, p. 1357-1366, NY, 1982

[6]. Carr, K.R., "An Evaluation of Industrial Platinum Resistance Thermometers," *Temperature, Its Measurement and Control in Science and Industry*, Instrument Society of America, Part 2, p. 971- 982, Pittsburgh,1972

[7]. Kerlin, T.W., and Shepard, R.L., " Industrial Temperature Measurement," Instrument Society of America, NC, 1982

[8]. Sandborn, V.A., "Resistance Temperature Transducers," Metrology Press, CO, 1972

[9]. Arendt, J.S., " Development of Methods For In-Situ Testing of Response Times of RTDs in Commercial Pressurized Water Reactors," *Thesis*, The Univeristy of Tennessee, Knoxville, TN, March 1977

[10]. Curtis, D.J., " Platinum Resistance Interpolation Standards," *Temperature, Its Measurement and Control in Science and Industry*, Instrument Society of America, Part 2, p. 951- 961, Pittsburgh,1972

[11]. Schooley, J.F., " Thermometry," CRC Press Inc., FL, 1986

[12]. Kutz, M., " Temperature Control," John Wiley & Sons Inc., NY, 1968

[13]. Kerlin, T.W., et al., " Temperature Sensor Response Characterization," *Electric Power Research Institute*, EPRI NP-1486, August 1977

[14]. Benedict, R.P., " Fundamentals of Temperature, Pressure, and Flow Measurements," John Wiley & Sons Inc., NY, 1977

[15]. Quinn, T.J., " Temperature," Academic Press Inc., NY, 1983

[16]. Warring, R.H., and Gibilisco, S., " Fundamentals of Transducers," Tab Books Inc., PA, 1985

[17]. Lucas, D.A., " Miniature Precision Platinum Resistance Thermometers," *Temperature, Its Measurement and Control in Science and Industry*, Instrument Society of America, Part 2, p. 963-969, Pittsburgh, 1972

[18]. Dutt, M., " Practical Applications of Platinum Resistance Sensors," *Temperature, Its Measurement and Control in Science and Industry*, Instrument Society of America, Part 2, p. 1013-1019, Pittsburgh, 1972

[19]. Caudill, M., Butler, C., " Understanding Neural Networks: Computer Explorations," Books' I and II, MIT,1990

[20]. Uhrig, R.E., *Private Communication*, The University of Tennessee, Knoxville, TN, December 1990

[21]. Maren, A., Pap, R., and Harston, C., " Handbook of Neural Computing Applications, " Academic Press Inc., CA 1990

[22]. E. Eryurek, B.R. Upadhyaya, "Sensor Validation For Power Plants Using Adaptive Backpropagation Neural Network," Department of Nuclear Engineering, The University of Tennessee, Knoxville, 1990.

[23]. Maureen Caudill, "Neural Networks Primer," *AI Expert*, June 1988.

[24]. "ANSim User's Manual, " Science Applications International Corp., CA, 1989.

[25]. "NeuralWorks, " NeuralWare Inc., PA, 1988.

# APPENDIXES

# APPENDIX A
# $R_S$ COMPUTATION

In a loop current step response (LCSR) test, a standard Wheatstone bridge has been modified. A resistor $R_s$ and a switch are added to the bridge as shown in Figure A.1. The $R_s$ and the switch are used to increase the current through the RTD from 1 mA, a normal-operation current, to 20 mA. The resistance of the $R_s$ needed is computed from the following calculations :

If the switch is closed, then

$$\frac{1}{R_{br}} = \frac{1}{R + R_{ad}} + \frac{1}{R + R_{TD}} = \frac{2R + R_{ad} + R_{TD}}{(R + R_{ad})(R + R_{TD})}$$

or

$$R_{br} = \frac{(R + R_{ad})(R + R_{TD})}{2R + R_{ad} + R_{TD}} , \qquad (A.1)$$

where

$R$ = the fixed resistor, 100 Ω

$R_{ad}$ = the adjustable resistor

$R_{TD}$ = the resistance of the RTD, 110 Ω

$R_{br}$ = the total resistance of the bridge.

When the bridge is in balance, $R_{ad} = R_{TD}$, hence $R_{br}$ is

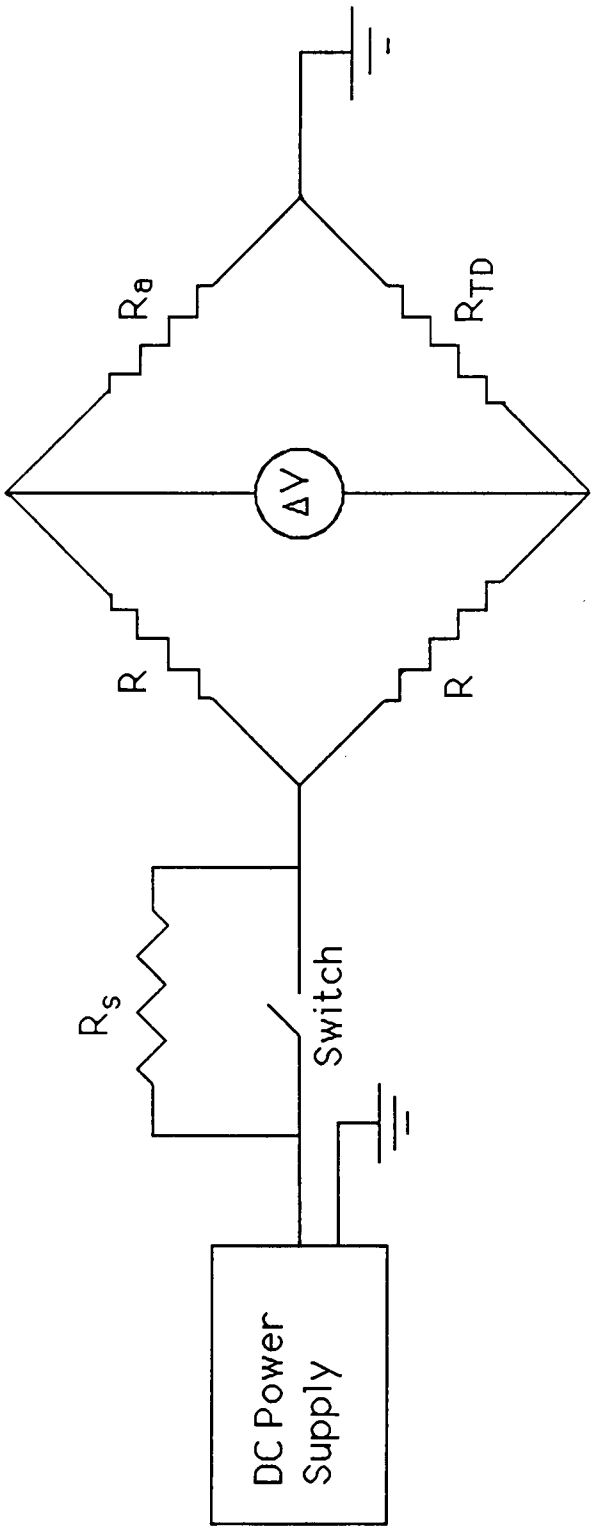



Figure A.1. Wheatstone Bridge Equipped With A Switch And Resistor $R_S$

$$R_{br} = 0.5\ (R + R_{ad}) = 0.5\ (100 + 110)\ \Omega = 105\ \Omega\ . \qquad (A.2)$$

The current through the RTD $I_2$ is 20 mA, so the voltage E needed is

$$E = I\ R_{br} = 2\ I_2\ R_{br} = (2)\ (20\ mA)\ (105\ \Omega) = 4.2\ V. \qquad (A.3)$$

If the switch is open, then the total resistance $R_{tot}$ of the circuit is

$$R_{tot} = R_s + R_{br} = (R_s + 105)\ \Omega \qquad (A.4)$$

and

$$E = I\ R_{tot} = 2\ I_2\ (R_s + 105)$$

or

$$R_s = (E/2I_2) - 105\ \Omega\ . \qquad (A.5)$$

Because E = 4.2 V and $I_2$ is 1 mA,

$$R_s = (4.2/0.001) - 105\ \Omega = 1995\ \Omega\ . \qquad (A.6)$$

However, the available resistor is 2025 Ω. If this resistor is used, then the current through the RTD at normal operation is

$$I_2 = E/[2(R_s + 105)] = 4.2/[2(2025 + 105)] = 0.99\ mA\ . \qquad (A.7)$$

It concludes that the resistance $R_s$ used is 2025 Ω.

## APPENDIX B

# PROGRAM FOR SELECTING LCSR DATA

The following program is written in FORTRAN language:

```
C *** This Program is for selecting LCSR data ***
C
      REAL X (500)
      INTEGER I
      OPEN (10, FILE='lcsr1.txt', STATUS='OLD')
      OPEN (20, FILE='20lc1.txt',STATUS='UNKNOWN')
      DO 100 I = 1, 500
            READ (10, 30) X(I)
100   CONTINUE
      DO 200 I = 1, 401, 20
            WRITE (20, 40) X(I)
200   CONTINUE
30    FORMAT (F16.6)
40    FORMAT (20(F8.4,','))
      CLOSE (10, STATUS='KEEP')
      CLOSE (20, STATUS='KEEP')
      STOP
      END
```

## APPENDIX C

# PROGRAM FOR SIMULATING NOISE

The following program is written in FORTRAN language:

```
C *** This Program is for simulating noise on LCSR data ***
C
      REAL X (500), RN1
      INTEGER I, J, K, ISEC, I100
      OPEN (10, FILE='I2j1a.txt', STATUS='OLD')
      OPEN (15, FILE='a.txt', STATUS='UNKNOWN')
      OPEN (20, FILE='20Ic1.txt',STATUS='UNKNOWN')
      CALL GETTIM (IHR, IMIN, ISEC*I100)
      RN_SEED = IHR + IMIN + ISEC*I100
      CALL SEED(RN_SEED)
      DO 100 I = 1, 500
            READ (10, 30) X(I)
100   CONTINUE
      DO 150 I = 1, 500
            CALL RANDOM(RN)
            RN = 2*RN - 1
            RN1 = 100*RN
            J = INT(RN1)
            K = MOD(J,2)
            IF (K.EQ.0) THEN
                  X(I) = 1.03*X(I)
                  ELSE
```

```
                X(I) = 0.97*X(I)
            ENDIF
            WRITE (15, 30) X(I)
150   CONTINUE
      CLOSE (15)
      OPEN (15, FILE='a.txt', STATUS='OLD')
      DO 170 I = 1, 500
            READ (15, 30) X(I)
170   CONTINUE
      DO 200 I = 20, 434, 7
            WRITE (20, 40) X(I)
200   CONTINUE
30    FORMAT (F16.6)
40    FORMAT (20(F8.4,','))
      CLOSE (10, STATUS='KEEP')
      CLOSE (20, STATUS='KEEP')
      STOP
      END
```

## APPENDIX D

# PROGRAMS OF SAMPLING INTERFACES

### 1. WDSAMPLE Program

```
**************************
*        MARSSA          *
*   PROGRAM WDSAMPLE     *
**************************
```

PROGRAM CREATES THE RAW DATA FILE FOR
THE ANALYSIS PROGRAMS. PROCESS SIGNALS
TO BE DIGITIZED ARE CONNECTED TO THE
A/D-BOARD

ENTER NUMBER OF FIRST AND LAST A/D-CHANNELS TO BE SAMPLED
(0. . . 15) :
ENTER NAME FOR SIGNAL NO. 1 :
ENTER GAIN FOR SIGNAL NO. 1 :

ENTER SAMPLING FREQUENCY (1.0 . . . 100 Hz) :
ENTER TOTAL TIME OF SAMPLING IN SECONDS
(MAXIMUM AVAILABLE SAMPLING TIME IS 3000 SEC.) :
ENTER GAIN FOR THE A/D-BOARD (1, 2, 4, 8) :
SAMPLING STARTS WHEN YOU PRESS RETURN

Pause.

Please press <return> to continue>

```
**********
* SAMPLING DATA ... *
**********
```

## 2. MATCON Program

```
**********
*  PROGRAM MATCON  *
**********
```

PROGRAM CONVERTS DATA FILES CREATED BY PROGRAM
WDSAMPLE OF MARSSA TO MATLAB FORMAT

ENTER THE NAME OF THE DATA FILE TO BE CONVERTED

*** INFORMATION ON THE DATA FILE ***

THE NAME OF THE DATA FILE :
TYPE OF DATA :
NUMBER OF DIGITIZED SIGNALS :
NUMBER OF SAMPLES PER SIGNALS :
SAMPLING INTERVAL (SEC.) :
GAIN OF THE A/D-BOARD :

ENTER THE NAME OF THE DATA FILE TO BE CREATED :

ENTER 1 IF YOU WANT TO CONVERT ALL CHANNELS (0 OTHERWISE) :

ENTER NUMBER OF SAMPLES TO BE CONVERTED :

ENTER THE INDEX OF FIRST SAMPLE TO CONVERTED :

CONVERTING . . .

### 3. WTIMEPLT Program

```
*****************************
*          MARSSA           *
*      PROGRAM WTIMEPLT     *
*****************************
```

PROGRAM PLOTS THE RAW MEASUREMENT DATA
FROM A FILE CREATED BY MEASUREMENT PROGRAM
WDSAMPLE OR CONVERSION PROGRAM WDTRANS.
IT CALCULATES ALSO THE APD FUNCTION AND
VARIANCE, SKEWNESS AND FLATNESS FOR SIGNALS.

ENTER NAME OF THE DATA FILE TO BE PLOTTED :

*** INFORMATION ON THE DATA FILE ***

THE NAME OF THE DATA FILE :
TYPE OF DATA :
NUMBER OF DIGITIZED SIGNALS :
NUMBER OF SAMPLES PER SIGNALS :

SAMPLING INTERVAL (SEC.) :

GAIN OF THE A/D-BOARD :

SIGNAL NAMES AND GAINS :

1. LCSR 1.00

2. START 1.00

Pause- please enter a blank line (to continue) or a DOS command.

ENTER THE PARAMETERS FOR PLOTTING AND/OR APD CALCULATIONS

ENTER THE FIRST SAMPLE IN ANALYSIS (>= 1), AND NUMBER OF SAMPLES TO BE SKIPPPED PERIODICALLY (>=0) IN ANALYSIS:

ENTER NUMBER OF SAMPLES PER BLOCK (<=1000) :

THE MAXIMUM NUMBER OF BLOCK IS 4. ENTER NUMBER OF BLOCKS TO ANALYZE:

ENTER 1 IF YOU WANT THE PLOTS TO BE PRINTED (0 FOR SCREEN PLOT):

**** ANALYSIS PARAMETERS ****

NO. OF SAMPLES PER BLOCK =

FIRST SAMPLE TO BE PLOTTED =

PERIODICAL SKIP IN PLOTTING =

NUMBER OF DATA BLOCKS/SIGNAL TO PLOT =

TOTAL NUMBER OF SAMPLES PER SIGNAL =

SAMPLING INTERVAL (SEC.) =

NUMBER OF SIGNALS TO PLOT =

Pause- please enter a blank line (to continue) or a DOS command.

```
**********************************
*      CALCULATING MOMENTS. . .  *
**********************************
```

ENTER 1 TO PLOT TIME SERIES PLOTS :

```
************************
*      READING DATA    *
************************
```

# VITA

Agus Cahyono was born in Semarang, Indonesia on November 15 1963. He entered Gadjah Mada University, Indonesia in July, 1982, and received the Engineer degree in nuclear engineering in June, 1987.

The following September he started working for Indonesian Atomic Energy Agency. Sponsored by Indonesian government, Agus continued his study in nuclear engineering at The University of Tennessee, Knoxville, in January, 1990. He received the Master of Science degree in December, 1991.